Intelligent System for Worm Detection

Ibrahim A. Farag
Faculty of Computers
and Information
Cairo University
Egypt

Mohammed A. Shouman
Faculty of Computers
and Information,
Zagazig University
Egypt

Tarek S. Sobh
Information Systems
Department
Egyptian Armed Forces
Egypt

Heba Z. El-Fiqi
Faculty of Computers
and Information
Zagazig University
Egypt

Abstract *Worms are on the top of malware threats attacking computer system although of the evolution of worms detection techniques. Early detection of unknown worms is still a problem. This paper produce a method for detecting unknown worms based on local victim information. The proposed system uses Artificial Neural Network (ANN) for classifying worm/ nonworm traffic and predicting the percentage of infection in the infected network. This prediction can be used to support decision making process for network administrator to respond quickly to worm propagation in an accurate procedure.*

Keywords*: Internet Worms, Worm Detection, Artificial Neural Network, Local Victim Information, Network Security.*

1. Introduction

The computer worm, which is a self-propagating malicious code, spread themselves without any human interaction and launches the most destructive attacks against computer networks. Li [12] describes the worm's life as consisting of many phases: target finding, transferring, activation, and infection. The first two phases cause network activities, worm behaviors in these two phases are critical for developing detection algorithms. This paper produces an artificial intelligence system for worm detection, that can detect worm virus in network with accuracy of %99.96 .Also this system can predict the percentage of worm infection in the network with absolute error average from 0% to 4%. Section 2 explores the dangerous of worm viruses and the current techniques which is used in worm detection. Section 3 describes the Artificial Intelligence (AI) used in worm detection. Section 4 produces the proposed model. Section 5 describes the implementation of the model. Section 6 presents the results of the system.

2. Network Attacks and Worm Detection

By studying reports produced by Trend Micro two years ago, the worms still one of the most infection malware codes dangerous. Eight of the top twenty threats counts of 2006 are worms [23]. Worm_NYXEM.E is on the top of this list. In the top twenty malware report from Dec 2006 to May 2007 [21], nine worms made it into the top twenty, reflecting the attractiveness of speedy propagation to malware authors. The first and the second place in the top ten detections in Trend Micro 2007 malware trends report [22] are worms. Wang [19] produces a system for early detection of worm based on ICMP collecting. It records the number of some type of data packets in detection interval. In detecting abnormal events, the system takes response actions and block susceptible IP and port. A router-based system to identify worm attacks by computing entropy values of selected packet attributes is presented by Zhou [7]. Another model is developed by Weaver [20]. This containment algorithm is suitable for the deployment in high-speed, low-cost network hardware, which can stop a scanning host after fewer than 10 scans with a very low false-positive rate. Kim [11] produces an algorithm to reduces both sensitivity and false alarm with weighted average queue length that smoothes sudden traffic changes. The algorithm counts the number of connection requests with different destinations, in contrast to simple length of delay queue as in the typical throttling algorithm. The queue length measuring strategy also helps reduce worm detection time and false alarm. Another method is detecting large-scale worm attacks using only end-host detectors.

2.1 Predicting Percentage of Infection

Several approaches are produced attempting to estimate the damage and predict the spread of worms; Kephart and White [10] developed The Epidemiological model, which is a simple model that explains the spread of computer viruses by employing biological epidemiology. The number of infected hosts depends on vulnerability density and scanning rate. The two-factor worm model by Zou et al [24], describes the behaviour of worm which based on two factors, the dynamic countermeasure by ISPs and users, and a slowed down worm infection rate. This model explains observed data for Code Red and the decrease in scanning attempts during the last several hours before it ceased propagation. The Analytical Active Worm Propagation (AAWP) model by Chen [2] extends the model of worms that employ random scanning to cover local subnet scanning worms. Parameters in this model include the number of vulnerable machines, size of hit lists, scanning rate, death rate, and patching rate. AAWP better models the behaviour of Code Red II than previous models. An approach to minimize the damage due to worm infection in enterprise networks which are produced by Sanguanpong [15] does not require observing variables during attacks. Therefore, it can be used to predict worm damage before the attack occurs. The result produced by Sanguanpong [15] has accuracy ranged from 83.33% to 90.91%, and False-Positive error rate of 0% to 4.16%

2.2 Behavioural Detection vs. Signature-Based Detection

Signature-based detection has been the first technique used to fight malware and still remains at the heart of nowadays antivirus software. Jacob [8] describes that these detection techniques search system objects such as files for suspicious byte patterns referenced in a base of signatures. Signatures can precisely identify the threat and name it; signature-based techniques are bound to detect known malware or trivial variants. But signatures are no longer simple byte patterns but complex meta-structures carrying dynamic aspects and a semantic interpretation. On the other hand, behavioral detection is thus more generic and more resilient to modifications than form-based detection.

2.3 Local Victim Information

Zou, Gao [25], and Staniford, [16] tried to explore global strategies techniques but it require a large monitored network (say, 220 nodes) to distinguish worms from other scanning activities. Some of them look to make nation-wide Internet worm control authority, others proposed to deploy sensors around the Internet. Although there is a need to global co-ordination to protect the Internet from worm intrusions, global detection strategies don't produce complete solution. Dagon and Xinzhou [3] discuss the idea of that since global detection strategies require large amounts of sensor data before detecting worm outbreaks, some local networks might be infected before learning about a worm outbreak. In global detection strategies, in order to gain sufficient worm traffic to become detectable, these strategies have to wait a lot of local networks to fall as victim to the worm. Other Researchers like Guofei [6] uses the idea of using distributed system that detects worm probing traffic through local traffic observations . From local networks point of view, it is more useful to know which machines are infected and how the attack is progressed. Thus worm detection techniques for smaller local networks needs more research.

3. Intelligence Techniques Used In Detecting Network Attacks

A recent survey of intrusion detection [9] suggests using artificial intelligence (AI) techniques to recognize malicious software (malware) in single computers and in computer networks. It describes the research done in developing these AI techniques, and discusses their advantages and limitations. Moskovitch [14] used machine learning techniques in classification of a computer behavior into malicious and benign. He focuses on the feasibility of accurately detecting unknown worm activity in individual computers while minimizing the required set of features collected from the monitored computer. Four feature selection methods were used to reduce the number of features and four learning algorithms were applied on the resulting feature subsets; four commonly used Machine Learning algorithms: Decision Trees, Naive Bayes, Bayesian Networks and Artificial Neural Networks. The evaluation results suggest that by using classification algorithms applied on only 20 features, the mean detection accuracy exceeded 90%, and for specific unknown worms accuracy reached above 99%, while maintaining a low level of false positive rate. Andrzej Bielecki [1] developed a neural approach to worm detection designed as a part of a multi-agent system intended to manage IP networks. The efficiency of virus recognition is about 95%. One of the AI techniques mentioned in that survey [9] is ANN.

3.1 Using Ann In Worm Detection

Stopel et-al [17] produced an approach for detecting the presence of computer worms based on ANN using the computer's behavioral measures. Stopel et-al [17] compared three different feature selection techniques for the dimensionality reduction and identification of the most prominent features to capture efficiently the computer behavior in the context of worm activity. In order to evaluate the different techniques, several computers were infected with five different worms and 323 different features of the infected computers were measured.

They evaluated each technique by preprocessing the dataset according to each one and training the ANN model with the preprocessed data. They then evaluated the ability of the model to detect the presence of a new computer worm, in particular, during heavy user activity on the infected computers.

Another research produced by Stopel et-al [18] used ANN and two other known classifications techniques, Decision Tree and k-Nearest Neighbors, to test their ability to classify correctly the presence, and the type of the computer worms even during heavy user activity on the infected computers. By comparing these three approaches, the ANN approach has computational advantages when real-time computation is needed, and has the potential to detect previously unknown worms. Also, ANN may be used to identify the most relevant, measurable features, and thus reduce the feature dimensionality. The model proposed by Stopel et-al [17, 18] detect malicious activity of worms by looking at the attributes derived from the computer operation parameters such as memory usage, CPU usage, traffic activity etc. The main drawback of this model appears in misclassifications that it still faces difficulties related to the detection of the worms in the beginning of their activity. Stopel et-al described main advantages of using ANN appears in worm detection as the high level of accuracy in real-time operation, low CPU resources utilization during the classification phase, and the ability to generalize, in order to detect and identify, any previously unseen classes. Bielecki and Hajto [1] present a neural-based agent for IP traffic scanning and worm detection. This approach of worm detection is designed as a part of a multi-agent system intended to manage IP networks. The efficiency of virus recognition is about 95%.

3.2 Using AI in Predicting Percentage of Infection

An approach to minimize the damage due to worm infection in enterprise networks is produced by Sanguanpong [15]. This approach did not require observing variables during attacks. Therefore, it can be used to predict worm damage, before the attack occurs. The model does not rely on attack type and configuration of the worm program. Such factors are: (1) Scanning rate in the Epidemiological and the AAWP model [2]. (2) Size of hit lists in the AAWP model. The prediction rate of the model produced by Sanguanpong [15] ranges from 83.33% to 90.91%. The false positive rate ranges from 0% to 4.16%.

According to the results of the research, Artificial intelligence adds precision to the new worms detection techniques. Also, ANN produced good results in real time operation and the ability to detect worms that it didn’t been trained.

4. The Proposed Model

This model is used to identify worm traffic from normal traffic; also it can predict the infection percentage in the network, which can be used by the administrator to take the appropriate action. This model depends only on the data that collected from the local victim information. As seen in Fig. 1, this model consists of 4 modules: (1) Traffic Statistical Analyzer Module (TSAM) (2) Port Matching Module (PMM) (3) Artificial Neural Network Module (ANNM) (4) Response Module (RM). The system works as follow: The incoming and outgoing traffic are monitored using sniffing tool. This traffic is used by TSAM to calculate some statistics. This monitored traffic is used as input to the PMM, which use the idea of infection-like-behaviour in worm spreading to identify suspected worm traffic. Then administrators apply the number of hosts online as an input to ANNM, which uses the data that collected from other modules to classify the traffic into worm traffic or normal traffic, and to predict the percentage of infection in the network.

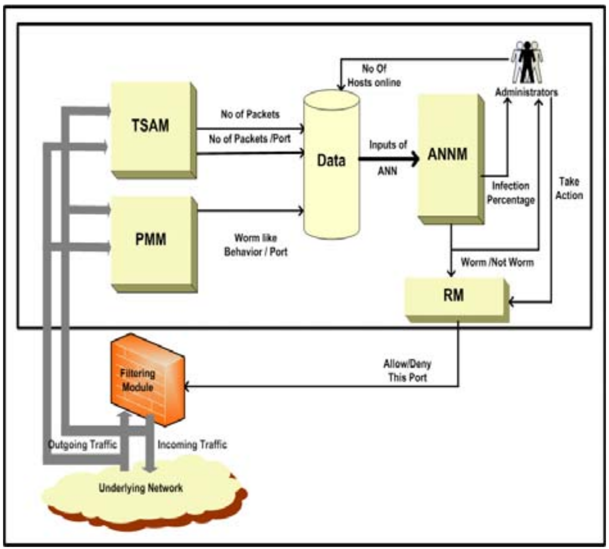

Figure 1.. The proposed model used for detecting worm behavior using ANN.

The administrator can uses ANN outputs to take the appropriate action to protect the network, and to alarm other subsystems, or partners about the spreading of the detected worm virus. Until administrator can apply the solution based on the company policy. The RM can be used to block the traffic on the suspected port(s) used in propagation of the detected worm. This action is used to prevent the propagation of the worm in the network, but it cannot heal the infected nodes.

4.1 Traffic Statistical Analyzer Module (TSAM)

This module is responsible for calculating statistical values based on the analysis of incoming and outgoing traffic. It captures the traffic and calculates the number of packets per time unit, number of packets produced by each source/destination port in time unit, also it can produce number of packets per protocol in a time unit. But only number of packets and number of packets per port that are used as input to the data set for ANN.

4.2 Port Matching Module (PMM)

Being fully automated, a worm's behaviors usually repetitious and predictable, making it possible to be detected. Guofei, Monirul et al [6] states that "After a vulnerable host is infected by a worm on a port I (i.e., the host is the destination of an early worm attack), the infected host will send out scans to other hosts targeting at the same port I in a short time". This module shown in Fig.2 uses this idea to produce the number of packets per port that match the worm infection behaviour. Since there is no way to know if a packet source is victim or slave attacker; so each record is being examined as if it is from the victim or from slave attacker. Then in a selected unified time interval, if a packet is sent from a slave to a victim on specific port, followed by a packet is sent from this victim IP address to the same destination port, thus is counted as worm-like behavior on that port. A dynamic table is made to produce number of occurrence for this worm like behavior per each used port.

4.3 Artificial Neural Network Module (ANNM)

A supervised ANN can be trained to take the values that represent the current behavior of the network under non-worm traffic and worm traffic. After sufficient number of iterations, it can be used as a control unit in the proposed system to identify the worm traffic.

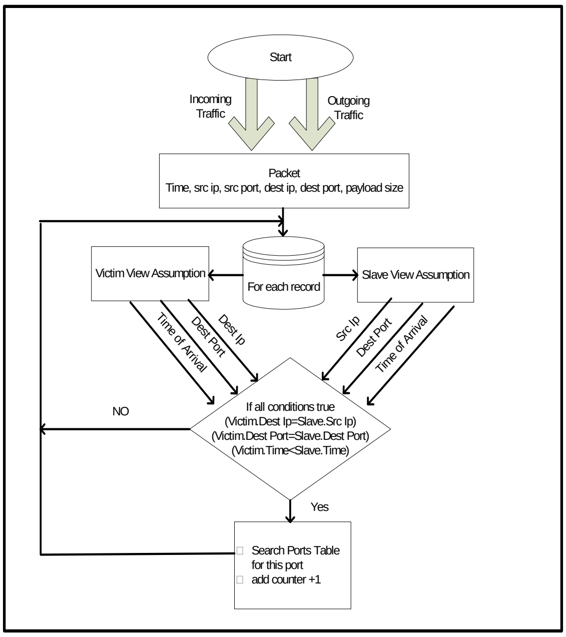

Figure 2.. Flowchart of PMM.

During the designing phase: One ANN was used to produce the classification and the prediction problem, this is proposed in CPC Model. To enhance the results, the two problems is separated into two ANNs. Each one has the objective to solve certain problem; this idea is proposed in the CPS model.

4.4 Classification / Prediction Combined Model (CPC Model)

In this model, shown in Fig. 3, the idea was to use one ANN to produce the two desired outputs.

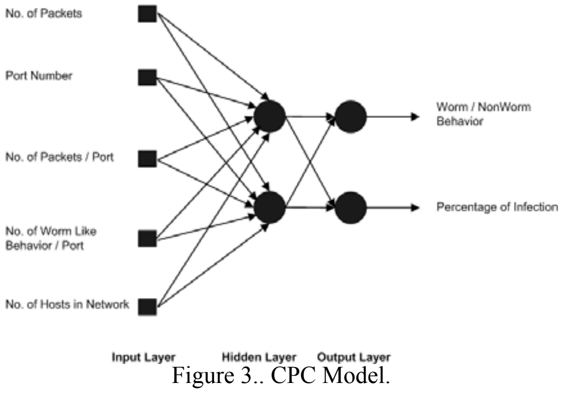

Figure 3.. CPC Model.

4.5 Classification / Prediction Separated Model (CPS Model)

In this model, two ANN networks are used: First ANN in Fig. 4-a is used to solve the classification problem. This ANN produces two outputs: worm behavior class, and normal behavior class. The result is producing to any class the traffic belongs. Second ANN, which produced in Fig., 4-b, is used to solve prediction problem This ANN produces one output: percentage of infection in the network.

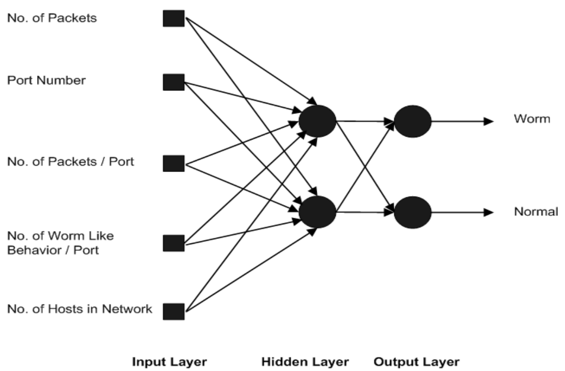

Figure 4-a. CPS Model (Classification).

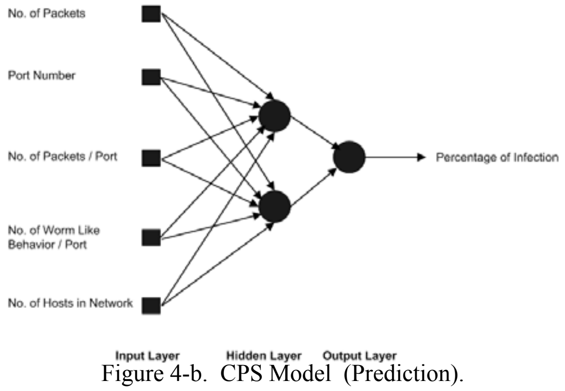

Figure 4-b. CPS Model (Prediction).

4.6 Response Module

This module is responsible for applying the action recommended by the administrator. It can be designed to take automatic action. Its objective is to reconfigure the packet firewall to block traffic on the suspected port(s) that is used in worm propagation. Administrator can then take an appropriate action based on the company/ organization security policy. Using this system, administrator knows if the monitored network is infected or not, and in case of infection the percentage of infection.

5. Implementation

The implementation of this model is done for all the modules, except the response module; which will be implemented in future work.

5.1 GTNetS Simulator

Network simulations are used to study the worm propagation. Sharif [5] built worm models into GTNets [4] and was able to simulate networks having hundreds of thousands of hosts and measure the effect of different network parameters on the worm spread rate. Liljenstam et al [13] used the SSFNet simulator with packet-level details for a small section of the network and represented the rest of the Internet with an analytic model. In this system the worm models in GTNets is used, which model the behavior of scanning worms [5].

5.2 Developing ANNM

Data are collected using a developed application called (Worm Detection Traffic Analyzer). Input Data includes 5 features elected from many features that help in identifying worm behavior from non-worm network behavior. All of these inputs are numerical values. The normalization of value from 0 to 1 is done by the tool used for building the ANN model which is NeuroSolution.

The Dataset consists of 5430 exemplars: 4230 examplers are used for training (% 78), 1200 examplers are used for testing (% 22). Collected data are gathered from different experiments, each experiment produces many exemplars, and the sequence within the same experiment is matter. When this application is used in the real life application, it means that the sequence of traffic is producing information in the network traffic; data are related in this way. But every experiment is completely separated from other experiments. Then experiments orders are randomized to accomplish different conditions randomly, but within the same experiments result, data can’t be randomized.

After testing different topologies, the best result was chosen to produce the model. The Network Training Paradigm is Supervised Learning implemented using Multilayer Perceptron (MLP) which is feed forward neural network that uses Back propagation algorithm. After testing some activation functions, best results are produced by Sigmoid Function. Also Momentum at step size 100 produces the best results as learning rule

There are two methods for updating weights that can be used: On-Line, Batch learning. After testing both of them, best results are produced by online updating.

6. Results

The two proposed model are evaluated and compared by each other. The CPS Model has better results than the CPC Model.

6.1 Evaluation Parameters

In the field of artificial intelligence, a confusion matrix is a visualization tool typically used in supervised learning. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. For evaluation purposes, the True Positive Rate (TPR), which is the number of positive instances classified correctly and shown in Eq. (1), is measured, and the False Positive Rate (FPR), which is the number of negative instances misclassified and shown in Eq. (2), is measured. The Total Accuracy measures the number of absolutely correctly classified instances, either positive or negative, divided by the entire number of instances shown in Eq. (3). The absolute error of Prediction ε can be calculated using Eq. (4). Mean Squared Error (MSE) is the average of the square of the difference between the desired response and the actual system output (the error), the formula for the mean squared error is shown id Eq. (5).

$$TPR = \frac{|TP|}{|TP| + |FN|} \tag{1}$$

$$FPR = \frac{|FP|}{|FP| + |TN|} \tag{2}$$

$$Total \quad Accuracy = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|} \tag{3}$$

$$\varepsilon = | P - P^{*} | \tag{4}$$

$$MSE = \frac{\sum_{j=0}^{P} \sum_{i=0}^{N} (d_{ij} - y_{ij})^{2}}{N \; P} \tag{5}$$

Where P = number of output processing elements, N is number of exemplars in the data set, y_{ij} is network output for exemplars i at processing element j, d_{ij} is desired output for exemplars i at processing element j

6.2 Experiment I

Six tests are done by the CPC Model to test percentage of infection; one normal traffic test plus 5 different worms test, results are shown in Fig.5. In Fig. 5-a, the error in prediction in the case of normal traffic doesn't exceed 5.00E-04. Figures 5-b to 5-f show that the error in the prediction is low, sometimes it turns to zero as in Fig. 5-d.

6.3 Test CPC Model Using Simulation Code Red II Worm Like For 383 Seconds

The Parameters used in this test are: Network Size is 18000 nodes, Hosts online are 15266, Worm Type is TCP worm, TCP Threads is 6, Vulnerability is 1.0, Payload length is 3818 bytes, infection port is port 80, worm target vector: Local Preference Scan (For the same first byte, the probability is 0.5, for the same first two byte, the probability is 0.375, and probability 0.125 for random IP). MSE of test is 5.9E-05. Results are presented in Fig. 6.

6.4 Experiment II

The CPC Model is tested using 25 different worms. These 25 tests are done to find the trend and powerness, and weakness of the system. The test results are used to compare this model to the second proposed model.

For worm detection, TPR=99.10%, FPR= 0.0721%, Total accuracy=99.71%. The results of predicting percentage of infection are Mean of Error is 8.78E-02, Standard Deviation of Error is 3.47E-02, Minimum Absolute Error in Predicting Percentage of infection is 7.2E-05, and Maximum Absolute Error in Predicting Percentage of infection is 5.55E-01.

6.5 Experiment III

This experiment is a repetition of EXPERIMENT II using the CPS Model. For worm detection, TPR= 99.85%, FPR= 0%, Total Accuracy= 99.96%. The results of predicting percentage of infection are Mean of Error is 4.52E-03, Standard Deviation of Error is 2.37E-03, Minimum Absolute Error in Predicting Percentage of infection is 3.00E-06, and Maximum Absolute Error in Predicting Percentage of infection is 4.02E-02.

MSE of percentage of infection is 7.59E-04; MSE of classification is 8.15E-04. Fig. 7 shows the results of tests done with code red II repeated four times with the CPS Model.

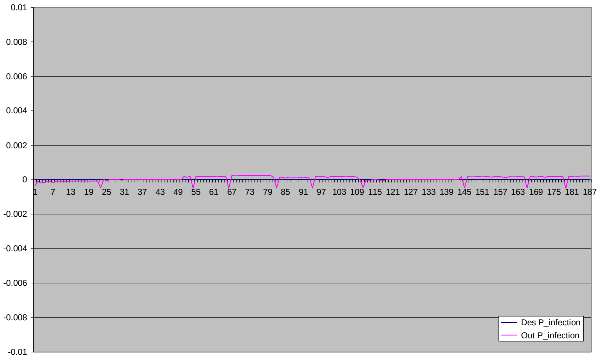

Figure 5-a. Prediction of infection percentage in normal traffic

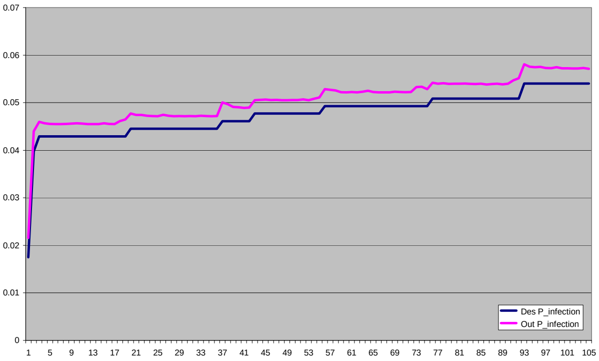

Figure 5-b. Prediction of infection percentage in worm 1 test.

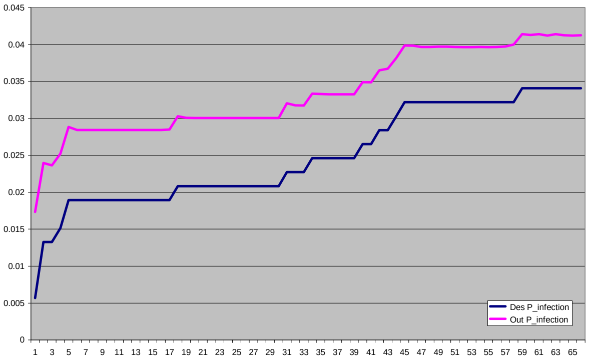

Figure 5-c. Prediction of infection percentage in worm 2 test.

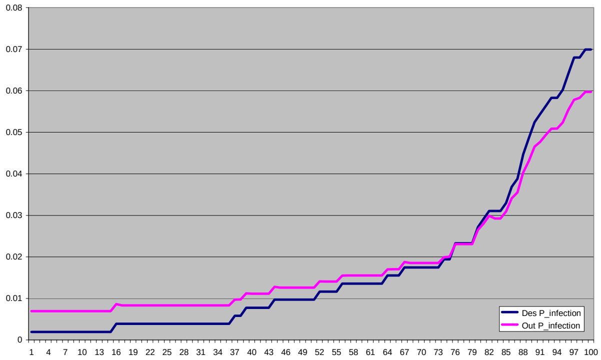

Figure 5-d. Prediction of infection percentage in worm 3 test.

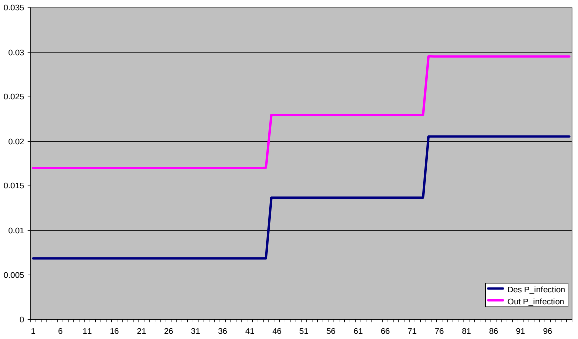

Figure 5-e. Prediction of infection percentage in worm 4 test.

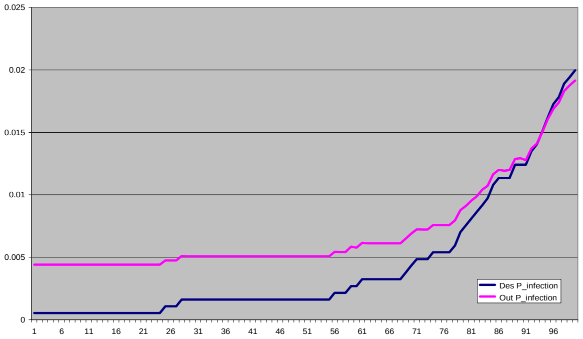

Figure 5-f. Prediction of infection percentage in worm 5 test.

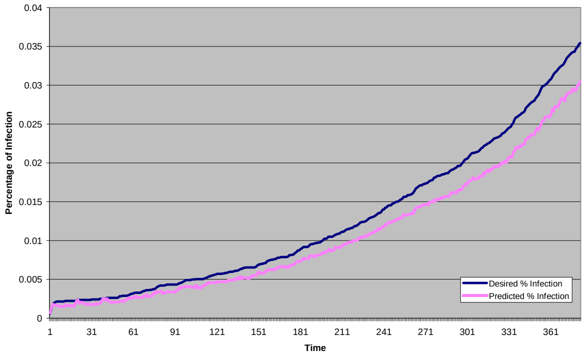

Figure 6. Prediction of percentage of code red II infection in the network by the CPC Model.

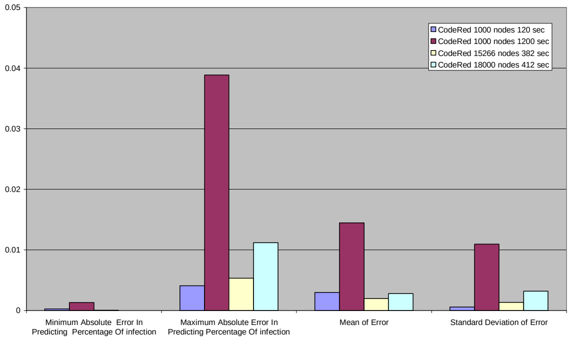

Figure 7. Comparison between four CodeRed II tests

6.6 Comparison Between The Two Models

The results of the comparison between two models are done by studying these models. The relation between the error in prediction and other parameters are evaluated. The result of this comparison can not be presented because the lack of space, so some of these results are shown here in Fig. 8. Tables 1 produce the TPR, FPR and the accuracy of the worm detecting in the two models. Although that the CPS Model has better performance than the CPC Model, the difference is not significant. The comparison between errors in prediction of infection between the two models which are shown in table 2.

Fig. 8-a show the relation between number of threads in TCP worms and error in predicting the percentage of infection in the network. Fig. 8-b shows the relation between scan rate in UDP worms and error in predicting the percentage of infection in the network. Fig. 8-c shows the relation between payload size of the worm and error in predicting the percentage of infection in the network. Fig. 8-d shows the relation between worm target vector and error in predicting the percentage of infection in the network. Fig 8-e shows the relation between number of online hosts and error in predicting the percentage of infection in the network. Fig 8-f shows the relation between worm scan range and error in predicting the percentage of infection in the network. All these comparisons indicate that the CPS Model has better performance.

Table 1. Comparison between accuracy of CPC and CPS

Experiment Number	I CPC Model	II CPC Model	III CPS Model
TPR	1	99.10%	99.85%
FPR	Undefined	0.07%	0%
Total accuracy	1	99.71%	99.96%

Table 2. Comparison between absolute errors in the experiments in CPC and CPS

	CPC Model	**CPS Model**
Mean of error	8.78E-02	4.52E-03
Standard deviation of error	3.47E-02	2.37E-03
Minimum absolute error in predicting percentage of infection	7.20E-05	3.00E-06
Maximum absolute error in predicting percentage of infection	5.55E-01	4.02E-02

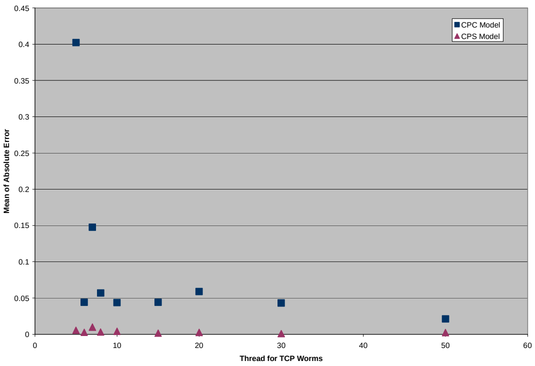

Figure 8-a. Relation between absolute error in infection percentage and threads for TCP worms.

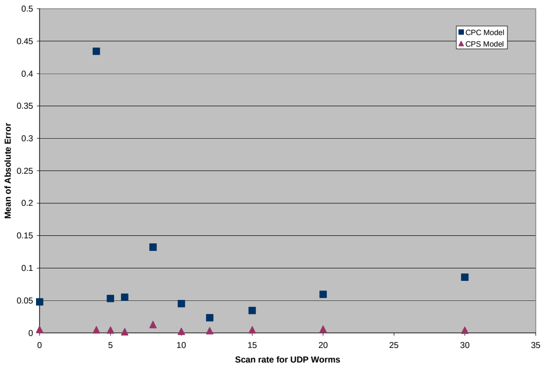

Figure8-b. Relation between absolute error in infection percentage and scan rate for UDP worms.

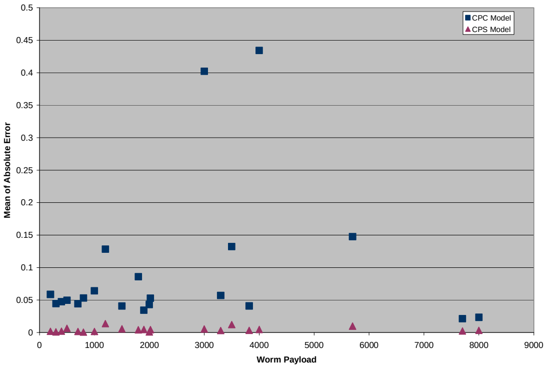

Figure 8- c. Relation between absolute error in infection percentage and worm payload.

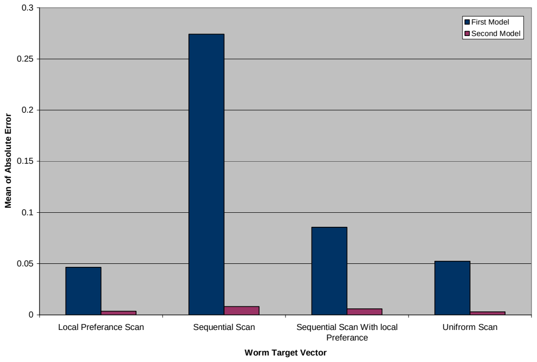

Figure 8-d Relation between absolute error in infection percentage and worm target vector.

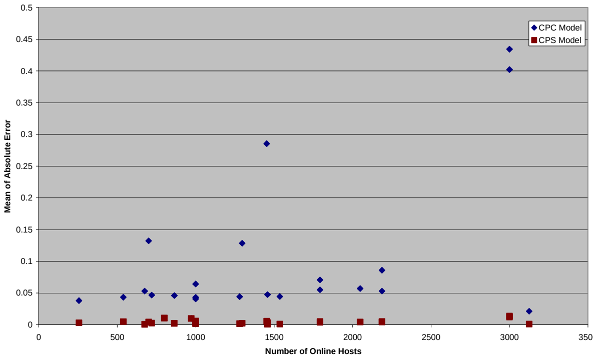

Figure 8-e Relationship between absolute error in infection percentage and number of online hosts.

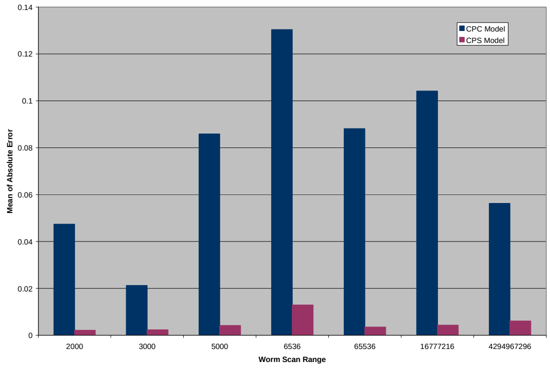

Figure 8-f Relationship between absolute error in infection percentage and worm scan range.

7. Conclusion

The proposed model produces good results in worm detection. The advantage of the ANN method over other techniques is its ability to classify correctly a worm not used in the training. The proposed system produces perfect result with accuracy of %99.96 in detecting the presence of worm in the network even for unknown worms.

The prediction of percentage of infection has better performance in the CPS Model than in the CPC Model. Average of the error in prediction is 4.52E-03. When testing system in different states. The system has good performance for both small and large network. The drawback of this system is that when the system become highly infected, the error in prediction increases slightly. Thus it is recommended in future work to modify the learning of ANN module to include more training set in for the conditions of high infection.

References

[1] Bielecki, A. and P. Hajto, *A Neural-Based Agent for IP Traffic Scanning and Worm Detection* , 2004

[2] Chen, Z., L. Gao, and K. Kwiat, "Modeling the Spread of Active Worms". *INFOCOM. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE* , 2003

[3] Dagon, D., et al., "Honeystat: Localworm Detection Using Honeypots" *, 2004*

[4] George, F.R., "The Georgia Tech Network Simulator". ACM , 2003

[5] George, R., S. Monirul, and L. Wenke, "Simulating Internet Worms". IEEE Computer Society , 2004

[6] Guofei, G., et al., "Worm Detection, Early Warning and Response Based on Local Victim Information". *Proceedings of the 20th Annual Computer Security Applications Conference*. Arizona, USA: IEEE Computer Society. pp. 136-145, 2004

[7] Hanxun, Z., W. Yingyou, and Z. Hong, "Detecting Early Worm Propagation Based on Entropy". ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007

[8] Jacob, G., H. Debar, and E. Filiol, "Behavioral Detection of Malware: From a Survey Towards an Established Taxonomy". *Journal in Computer Virology*, 2008.

[9] Kabiri, P. and A.A. Ghorbani, "Research on Intrusion Detection and Response: A Survey". *International Journal of Network Security*, vol. 1(2), pp. 84-102 , 2005.

[10] Kephart, J.O. and R.S. White, "Directed-Graph Epidemiological Models of Computer Virus Prevalence". *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 343-359, 1991.

[11] Kim, J., et al., *Reducing Worm Detection Time and False Alarm in Virus Throttling* , 2005

[12] Li, P., M. Salour, and X. Su, "A Survey of Internet Worm Detection and Containment". *Communications Surveys & Tutorials, IEEE*, vol. 10(1), 2008.

[13] Michael, L., et al., "Simulating Realistic Network Worm Traffic for Worm Warning System Design and Testing". ACM , 2003

[14] Robert Moskovitch, Y. Elovici, and L. Rokach, "Detection of Unknown Computer Worms Based on Behavioral Classification of the Host". *Computational Statistics and Data Analysis, ElSevier*,vol , 2008.

[15] Sanguanpong, S. and U. Kanlayasiri, "Worm Damage Minimization in Enterprise Networks". *International Journal of Human-Computer Studies*, vol. 65(1), pp. 3-16 , 2007.

[16] Staniford, S., V. Paxson, and N. Weaver, "How to Own the Internet in Your Spare Time" *, 2002*

[17] Stoppel, D., et al., "Improving Worm Detection with Artificial Neural Networks through Feature Selection and Temporal Analysis Techniques". *International Conference on Neural Networks ICNN* International Journal of Applied Mathematics and Computer Sciences, 2006

[18] Stoppel , D., et al., "Application of Artificial Neural Networks Techniques to Computer Worm Detection". *Neural Networks,. IJCNN. International Joint Conference*, 2006

[19] Wang, J., et al., "Internet Worm Early Detection and Response Mechanism". *The Journal of China Universities of Posts and Telecommunications*,vol. 14(3), 2007.

[20] Weaver, N., S. Staniford, and V. paxson, *Very Fast Containment of Scanning Worms, Revisited*, 2007

[21] Yaneza, J.L., *"1h 2007 Threat Roundup and 2h 2007 Forecast,"* Trend Micro, 2007.

[22] Yaneza, J.L., et al., *"2007 Threat Report | 2008 Threat and Technology Forecast,"* Trend Micro Corp, 2008.

[23] Yaneza, J.L., et al *"The Trend of Threats Today: 2006 Annual Roundup and 2007 Forecast,"* Trend Micro, 2006.

[24] Zou, C.C., W. Gong, and D. Towsley, "Code Red Worm Propagation Modeling and Analysis". ACM Press , 2002

[25] Zou, C.C., et al., "Monitoring and Early Warning for Internet Worms". ACM Press, 2003

Prof. Ibrahim Farag received his B.Sc. degree in mathematics from faculty of science, Cairo University in 1964. He received his PhD in computer science in 1976, from Manchester University, England. He is currently a full-time professor at faculty of computers and information, Cairo University. During his academic life, he achieved many positions such as chairman of computer science department, and dean of faculty of computers and information, Cairo University. During his academic career he published many papers in extendable programming language design, computer networks, and software engineering.

Prof. Mohamed Shouman is the Dean of Computers and Informatics College, Zagazig University, Egypt. He received his BS degree in Production Engineering, his MS degree in Production Engineering (A study of a general Time/Cost Relation in Network Analysis), and his PhD degree in Production Engineering (On the Optimization of Resource Usage in the Project Network of Cyclic Nature). His research interests include Industrial engineering, Mathematical modeling and computer applications, Network analysis, Project Management, Transportation problem, Implementation and design of simulation models, Inventory models and quality control, Scheduling techniques, CAD/CAM and CAPP techniques, Flexible manufacturing systems, Experimental design techniques, Expert systems and artificial intelligent techniques, Form error evaluation, Facility layout planning and design, Neural network, Intelligent information systems, Optimization techniques, Genetic Algorithms.

Dr. Tarek Salah Sobh received his B.Sc. degree in computer engineering from Military Technical College, Cairo, Egypt in 1987. Both M.Sc. and Ph.D. degrees from Computer and System Engineering Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt. He has managed, designed and developed several package for business applications and security systems. He has authored/co-authored of many refereed journal/ conference papers and booklet. Some of the articles are available in the ScienceDirect Top 25 hottest articles. His research of interest includes computer networks, security systems, distributed systems, knowledge discovery, data mining, and software engineering

Heba Zaki El-Fiqi is a Teaching Assistant in IT Dept, Faculty of Computers & Informatics, Zagazig University. She received her B.Sc. degree in 2003 from Computer Science Department, Faculty of Computers & Informatics, Zagazig University, Egypt. She received her M.Sc. degree in 2008 from Computer Science Department, Faculty of Computers & Information, Cairo University, Egypt. She is a member of Software Engineers Association (ESEA). Also, she is certified from Cisco as *Cisco Certified Network Associated* and as *Cisco Academy Instructor* and certified from Microsoft as *Microsoft System Engineer + Security*. Her research of interest includes computer networks, security systems, and intelligent systems.