Predicate based Algorithm for Malicious Web Page Detection using Genetic Fuzzy Systems and Support Vector Machine

S. Chitra[1], K.S. Jayanthan[2], S. Preetha[3] & R.N. Uma Shankar[4]

[1,2,3,4]CIET,
Coimbatore, TN, India.

ABSTRACT

In the era of internet, users are keen to discover more in the web. As the number of web pages increases day-by-day malicious web pages are also increasing proportionally. This paper focus on detecting maliciousness in a web page using genetically evolved fuzzy rules. The above formed rules are filtered by Support Vector Machine and finally storing the result in a symbolic knowledge base, with appropriate weightage for each rule. This provides an insight to symbolic and non-symbolic intelligence in malicious web page detection.

General Terms

Malicious web page, Static features, Potential features, Rule weightage.

Keywords

Malicious web page, Genetic fuzzy system, prolog, Support vector Machine.

1. INTRODUCTION

Web page requires largest security concern in the Internet. They are grouped to form a web site with numerous links and hyperlinks. Web pages may contain multimedia, images, videos and text. A web site is a collection of documents which can be accessed using internet. Malicious web page contains potential threats, which is a collection of scripts, foreign contents or an exploited content, added by an intruder to a web site. While performing a web service, a user will make request for a particular web site to a web server. The request is sent using communication messages. The server responds to the request by sending the requested web site to the user. It may contain malicious content or other software that install malware in the client's system. These are powerful to disclose the user's confidential informations that may include personal and professional matters without their knowledge.

The vulnerabilities of web browsers and web based services have increased numerously by out numbering the conventional security concerns in the computer society. Security concern should be made mandatory for every user while accessing a web page. In order to provide this security, the threat of maliciousness will be given more priority.

Illicit contents are advertised or made to display in the sites which are most commonly surfed. This is also found as spam or advertisements in electronic mails and is similar to 'click-jacking'. For unauthorized access of user passwords and account details, many phishing sites for reputed banks and other legitimate sites are available. Drive by-downloads are other type of techniques which is used to install or download malware in the client system that causes destruction of files and folders. It is possible by exploiting vulnerabilities of the web browser in the client side system.Machine learning approach is the mostly used method, which contains different concepts, related to malicious pattern detection. Features of web pages and changes in the system registries are selected to check this maliciousness. This paper proposes a method for detecting maliciousness using genetically evolved fuzzy rules. The potential features of a web page are extracted in this paper for implementation.

2. LITERATURE REVIEW

To detect malicious web pages, the approaches used are mainly classified as: State change technique, Signature technique and Machine learning approach. The major techniques used by an attacker on a system are SQL injection, Obfuscation, Browser vulnerability Exploitation and URL redirection. The popular search engines are being abused by compelling the users to visit malicious web sites.

State change approach which is also known as rule-based approach monitors change of state against unauthorized creation of executable files or register entries in the client system[33].To detect drive by-download attack Bragin *et al* used event triggers by creating some trigger conditions to find unauthorized activities in file system, process creation and registry system[7]. Behaviour monitoring module is also conducted in a client system to track malicious behaviour [24].SQL injection exploits a vulnerable database application that runs in web servers. It allows unauthorized operations in vulnerable databases to collect user information and also to make changes in the data itself. This allows the adversary to directly alter the contents of the server's database and inject his own content [23].

Signature system uses low or high interaction client side Honeypot. It uses signatures for finding the malicious web page. In HoneyC system, Snort signature is used and in Monkey-spider system, contents of web page are crawled and stored in files [1]. Unknown attacks are not considered in this approach which becomes its drawback. Machine learning approach consists of various methods to find maliciousness. HTTP responses from potential malicious web pages are analyzed to extract potential malicious characteristics [3], [22]. [14] proposed that malicious web pages usually change their display mode to be invisible or almost invisible. Another method proposes finding of malicious page by choosing features according to the usage of DHTML [34]. A Semantic aware reasoning detection algorithm based on structures of HTML codes is malicious web proposed to detect page [15]. Features of web pages, differentiated as static and run time are been selected to check the maliciousness of a web page using scoring algorithm [32]. Thus most of the machines learning approaches use features that are extracted from web page properties. Compromising a client system or a legitimate web site causes spread of malicious content. Thus this is not enough to classify malicious web page from a benign page.

3. METHODOLOGY

This paper proposes a method to find malicious web page using genetically evolved fuzzy rules. For this, potential features of a web page are selected. These features remain unchanged in a web page while it is executed. [32] Van Lam Le, Ian Welch, Xiaoying Gao, Peter Komisarczuk suggested 26 potential features from which we selected 21 optimal features.

Table 1. Potential Features

Sl.no.	Features
1.	Number of redirection
2.	Number of iframe and frame tag
3.	Number of external link in iframe and frame tag
4.	Iframe and frame link length
5.	Ratio of vowel character in iframe and iframe link
6.	Ratio of special character in iframe and frame link
7.	Number of external links(other than iframe)
8.	Other link length
9.	Number of scripts
10.	Number of script lines
11.	Ratio of special character in script
12.	Script length
13.	Script word length
14.	Script function argument length
15.	Number of objects
16.	Number of applets
17.	Object link length
18.	Ratio of special character in object links
19.	Ratio of vowel character in object links
20.	Number of object attributes
21.	Applet link length

Binary encoding and Rule creation are the two important steps in this approach. Genetic algorithm and fuzzy rules used to make the binary encoding of features and the rule creation respectively. The binary encoded conditions undergoes cross over to develop new and more combination rules. Thus various new rules of web page features can also be checked.

3.1 Genetic Fuzzy System

It is one of the most successful approaches to hybridize fuzzy systems with learning and adaption method apart from neural network [18], [19]. Genetic fuzzy systems are soft computing paradigm which focuses on the design and generation of fuzzy rules using evolutionary algorithm. It can solve complex real world problems which are difficult to be solved by conventional systems.

The Michigan-style genetic fuzzy rule-based system is a machine learning system which employs linguistic rules and fuzzy sets in its representation and is ideal for the rule discovery [12]. Genetic Algorithms are search algorithms based on natural genetics that provide robust search capabilities in complex spaces and thereby offer a valid approach to problems requiring efficient and effective search processes [8], [31]. This approach mainly used in all type of probabilistic optimization problems and is inspired by biological evolution process. A Genetic Algorithm maintains a population of candidate solution for the problem at hand, and makes it evolve by iteratively applying a set of stochastic operators. The operators mainly used are mutation and crossover.

A sample genetic algorithm shown below:

```
Produce an initial population
Evaluate the fitness of an individual
    while (true)
      Selection by Elitism
      Recombine individuals
      Apply mutation
      Evaluate the fitness
      Generate new population
    end while
```

3.2 Fuzzy System

A fuzzy rule based system consists of two components: knowledge base and inference system. Definition of the database and derivation of the rule base is associated with knowledge base. The definition of database is associated with variable universe, scaling factors of function, granularity per variable and membership functions associated with labels. Derivation of the rule base associated with fuzzy rule composition. The design of the knowledge base majorly focused on machine learning or human expert information.

3.2.1 Fuzzy logic

Fuzzy rules carries out the concept of partial truth or partial membership. There are membership values that range from 0.0 to 1.0 which denotes the false and truth respectively. Linguistic variables are essential to approximate the reasoning, because they are used to determine truth and possibility values of fuzzy propositions [21].Fuzzy logic provides an alternative way to represent linguistic and subjective attributes of the real world in computing. Fuzzy set operators are

(i) Union :

Union of $f_a(x)$ and $f_b(x)$ = max ($f_a(x)$,$f_b(x)$)

(ii) Intersection :

Intersection of $f_a(x)$ and $f_b(x)$ = min ($f_a(x)$,$f_b(x)$)

(iii) Complement :

Complement of $f_a(x)$ = 1- $f_a(x)$

Table 2. Membership Functions

Name	Membership function
Gaussian	$f_{gmf}\,(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}}$
Sigmoid	$f_{smf}\,(x;a,c) = \frac{1}{1+e^{-a\,(x-c)}}$
Triangular	$f(x;a,c) = max\left(min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right)$
Trapezoidal	$f(x;a,b,c,d)$ $= \; max\left(min\left(\frac{x-a}{b-a},1,\frac{c-x}{c-b}\right),0\right)$

3.4 Prolog

Kowalski, a logician, who showed logical proof, can support computation. Colmerauer and Warren contributed a significant role in developing prolog. It approximates first order logic. Prolog file contain a set of Horn clauses which contain facts and rules. It is a logic based language with few simple rules. The rule combines facts to increase the knowledge of the system. It is sometimes referred as 'symbolic logic'. The basic data structures used is tree. Variables are unknowns and not locations.

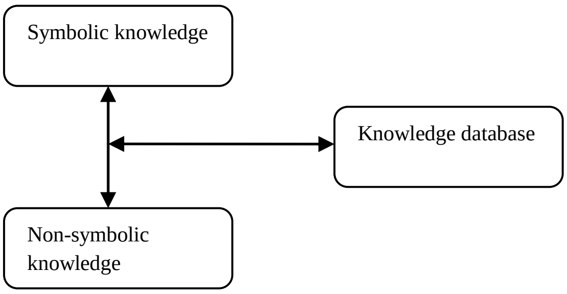

Figure 1. Knowledge flow

Prolog differs from other programming languages as it has no type. Prolog doesn't distinguish input and output but it solves relations and predicates. Unification is the built-in term manipulation method in prolog. The introduction of fuzzy logic into logic programming resulted in the development of several fuzzy prolog systems. These systems replaced the inference mechanism of prolog with a fuzzy variant which is able to handle partial truth [5]. The prolog execution environment deduces an answer from the relation definitions given.

3.5 Support Vector Machine

SVM is a bilinear classifier. It is introduced by Vapnik [31]. The major advantage of SVM is, it can introduce non-linearity into the classification process [4]. It uses Kernel trick for the above. It is trained using a set of features which make it easier to classify.

$$y_i(\omega^T x_i + b) \geq 1 - \; \varepsilon_i, \; i = 1,2,\cdots n \qquad (1)$$

where ω is the weight matrix

x is the input training vector

ε is the stack variable

To find an optimal hyper plane, is to solve the following constrained optimization problem.

$$Min_{\omega,\varepsilon} = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{N}\varepsilon_i \qquad (2)$$

subject to $y_i(\omega^T x_i + b) \geq 1 - \; \varepsilon_i$, where C is a user defined positive cost parameter and $\sum \varepsilon_i$ is an upper bound on the number of training errors. After solving (2), the final hyper plane decision function is achieved and

$$\begin{aligned} f(x) &= sign(\omega^T x + b) \\ &= sign(\textstyle\sum_{i=1}^{N} y_i\alpha_i(x,x_i) + \; b) \\ &= sign(\textstyle\sum_{i\epsilon SV} y_i\alpha_i(x,x_i) + \; b) \qquad (3) \end{aligned}$$

where α_i is a Lagrange multiplier and the training samples for which $\alpha_i \neq 0$ are support vectors (SVs).

The above linear SVM can be readily extended to a non-linear operator, Ø to map the input data into a higher dimensional feature space. In this way, it can solve non-linear problems. By replacing x in (1) and (2) with the feature space, $\emptyset(x)$ and solving the constrained optimization problem, the decision function,

$$\begin{aligned} f(x) &= \; sign\;(\textstyle\sum_{i=1}^{N} y_i\alpha_i(\emptyset(x),\emptyset(x_i)) + \; b) \\ &= \; sign\;(\textstyle\sum_{i=1}^{N} y_i\alpha_i k(x,x_i) + \; b) \\ &= \; sign\;(\textstyle\sum_{i\epsilon SV} y_i\alpha_i k(x,x_i) + \; b) \qquad (4) \end{aligned}$$

where $k(x_i,x_j) = \emptyset(x_i).\emptyset(x_j)$

Table 3. Kernel Functions

Kernel functions	Examples
Linear kernel	$k(x_i,x_j) = \; x_i^T x_j$
Polynomial kernel	$k(x_i,x_j) = (1 + x_i^T x_j)^p$
Gaussian kernel	$k(x_i,x_j) = \exp(-\frac{\lVert x_i - x_j \rVert^2}{2\sigma^2})$
Sigmoid	$k(x_i,x_j) = \; \tanh(\beta_0 x_i^T x_j \; + \; \beta_1)$

p (Best rule) = Rule with maximum weightage
= $w(r1)$ if $r_1 > r_2 \ldots\ldots r_n$

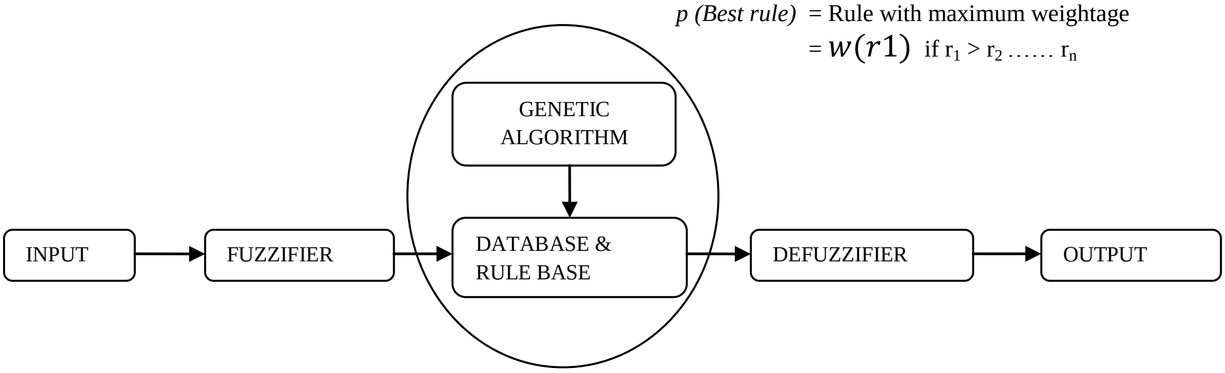

Figure 2: Genetic Fuzzy System

4. PROPOSED METHOD

Using Genetic Algorithm, from a set of initial Mamdani fuzzy rules the combinations of rules are created. This genetic algorithm itself will produce the combination of fuzzy rules using multiple-point mutation and double-point cross over. The method uses a probabilistic measure to apply genetic algorithm operators.

$$p\left({}^{mutation}/_{cross\ over}\right) = {}^{p(mutation\ \cap cross\ over)}/_{p(cross\ over)}$$

After completing rule creation, using Support Vector Machine classifier, the valid and invalid rules are filtered.

$$x\epsilon\ Fuzzy\ rule\ vector$$
$$y\epsilon + 1, -1$$

$$f(antecedence, consequence, \{valid, invalid\})$$

The SVM is trained using a set of manually created fuzzy rules with features as antecedence and consequences of the rule. This will produce non-linearity in the rule classification. Thus we are using radial basis function as kernel trick. The training pair consists of 125 valid and 125 invalid rules.

After the successful training, SVM is able to classify the genetically formed rules. The above formed filtered rules are stored in a symbolic knowledge base. Prolog is the best known symbolic knowledge base with well known interpretation capacity. For each webpage extract the basic feature vectors and calculate the antecedent occurrence frequency of the corresponding rules. That measure can be taken as the weightage of the rule and is given by

max (A/A+B, B/A+B)

A rule with ‘n’ features in its antecedent part, have a weightage which is given by

$$w(r) = max\left(\frac{A1}{\sum_{i=1}^{n} Ai}, \frac{A2}{\sum_{i=1}^{n} Ai}, \ldots \frac{An}{\sum_{i=1}^{n} Ai}\right)$$

$A1, A2, \ldots An$ are the feature vector frequencies in the webpage. Expectation of taking best rule,

$$r_1\, p_1 + r_2 p_2 + r_3 p_3 + \ldots + r_n p_n = 1$$

4.1 Algorithm

1. Start

2. Assign the Predicate variables in the Symbolic Knowledge base to null.

   Predicate variable, r = NULL

3. Initialize the fuzzy rules using MAMDANI fuzzy model.

   If Ai && Aj then Ri

   If Bi && Bj then Rj

   If Ci && Cj then Rk

4. Generate new featured fuzzy rules using Genetic Algorithm for a fixed number of iterations, using mutation and cross over.

   A’i && B’j then R’j

   A’j && B’i then R’i

   A’i && C’i then R’k

   B’i && C’j then R’i

5. Using a Support Vector Machine to classify the evolved rules as valid or invalid.

   Rule (r) =p(y=1| f*)*
   where f* is the feature vector
   P(y=1) is the probability of getting valid rule.

6. Store the above formed rules in Symbolic Knowledgebase.

   Malicious (A,B, Truth value)

7. Dynamically assign weightage to each symbolic rule using web feature frequency of the antecedent part.

$$w(r) = max\left(\frac{A1}{\sum_{i=1}^{n} Ai}, \frac{A2}{\sum_{i=1}^{n} Ai}, \dots \frac{An}{\sum_{i=1}^{n} Ai}\right)$$

8. Apply the above formed weighted rules on the features of the webpage to confirm the maliciousness.
9. Stop.

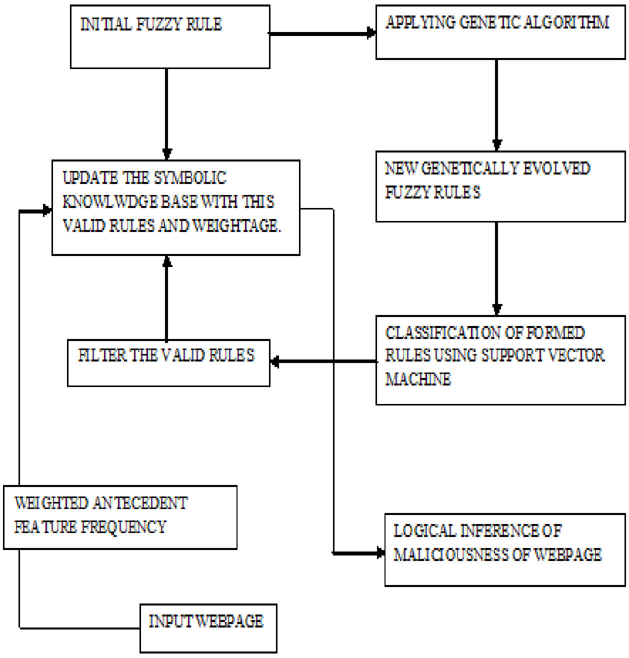
INITIAL FUZZY RULE
APPLYING GENETIC ALGORITHM
UPDATE THE SYMBOLIC KNOWLWDGE BASE WITH THIS VALID RULES AND WEIGHTAGE.
NEW GENETICALLY EVOLVED FUZZY RULES
FILTER THE VALID RULES
CLASSIFICATION OF FORMED RULES USING SUPPORT VECTOR MACHINE
WEIGHTED ANTECEDENT FEATURE FREQUENCY
LOGICAL INFERENCE OF MALICIOUSNESS OF WEBPAGE
INPUT WEBPAGE

Figure 3. Architecture of Malicious Detection System

5. IMPLEMENTATION

The proposed method is simulated using matlab and libSVM. The genetic algorithm tool kit in matlab generates fuzzy rules. Thus formed rules are classified into valid and invalid using support vector machine. This SVM is trained using manually created fuzzy rules and its antecedence and consequences as features. LibSVM tool kit is used for this purpose. The valid rules are stored in a symbolic knowledge base and assigns weight to each rule. Prologue is the knowledge used for this. Using MATLAB and LIBSVM implemented the basic concept of malicious detection. The implemented approach has more advantages than existing systems.

6. RESULT AND DISCUSSION

Sample formed rules:

If(number of objects= low) or (number of applets = high) then malicious= high

If (number of objects = medium) or (number of applets = high) then malicious = high

If (number of objects = low) or (number of applets = medium) then malicious = medium

If (number of objects = low) or (number of applets = low) then malicious = low

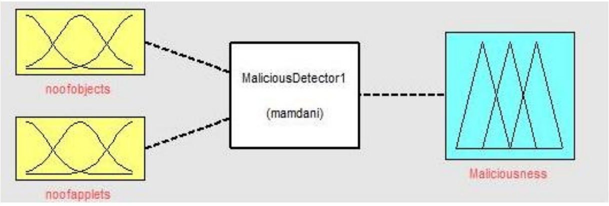

Figure 4. Fuzzy controller for Malicious Detector

The formed rules are converted into bit form in the ollowing way. As there are 26 features, the feature vector is represented as 5 bit vector. Linguistic variables are represented using 2 bits. Output variable is represented using 1 bit. Thus the single rule will be of 17 bits that includes two features and result. The SVM is trained with features as manually created fuzzy rules.

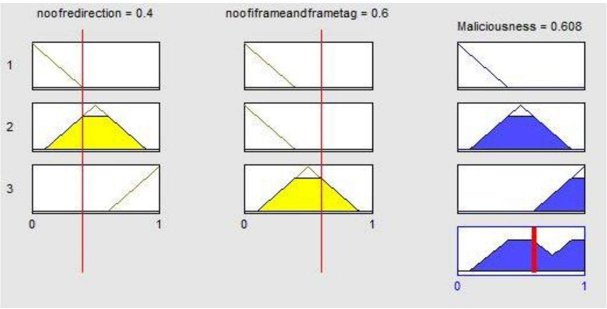

Figure 5. Fuzzy rule with membership value

Symbolic representation of knowledge:

$$Malicious\ (X, Y, truthvalue).$$

For each rule we have to allot appropriate weightage using *max (A/A+B, B/A+B)*. Then the probability of selecting best rule is pointed to the rule with maximum weightage.

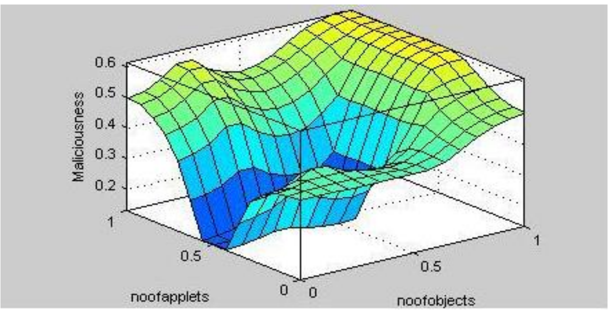

Figure 6. Fuzzy rule surface

The best rule thus selected will work in SVM to classify valid and invalid. We trained 2453 instances of web pages with 900 malicious web pages and our algorithm found out almost 853 malicious web pages with an accuracy of approximately 95.6%.

Table 4. Efficiency of SVM with kernel RBF

Trained rules for SVM in pair(positive, negative)	Efficiency
50, 50	91.2%
100 , 100	93.1%
125 , 125	93.5%

7. CONCLUSION AND FUTURE WORK

The proposed system has more advantage than existing system as it integrates symbolic intelligence with non symbolic intelligence .The formed rules accurately measures the maliciousness of the web page. This work majorly concentrates on malicious detection using genetically evolved fuzzy rules. The evolved fuzzy rules are classified as valid or invalid using SVM. Valid rules are rules then stored in symbolic knowledge base then to find out the efficient rule using Max-weight technique. The trained support vector machine can make the performance of the rules, better. Inclusion of more number of features and typeII fuzzy are further enhancement for this work.

8. REFERENCES

[1] A.Ikinci, T. Holz and F. Freiling, *Monkey-Spider: Detecting Malicious Websites with Low-Interaction Honeyclients*, *Sicherheit*, Saarbruecken, 2008.

[2] C. Seifert, I. Welch and P. Komisarczuk, *HoneyC - The Low- Interaction Client Honeypot, NZCSRSC*, Hamilton, 2007.

[3] C. Seifert, I. Welch and P. Komisarczuk, *Identification of Malicious Web Pages with Static Heuristics*, Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian, 2008.

[4] Chia-FengJuang, Shih-Hsuan Chin and Shu-Wew Chang, *A self organizing TS-Type Fuzzy Network with Support Vector Learning and its Application to Classification problems,* IEEE transactions on Fuzzy Systems, vol. 15, no.5, 2007.

[5] Claudio Vaucheret, Sergio Gaudarrama and Susana Munoz, Fuzzy prolog :a simple general implementation using CLP(R).

[6] Davis, La Jolla, CA: Morgan Kaufmann, *Adapting operator probabilities in genetic algorithms*, Proceedings of the Third International Conference on Genetic Algorithms, 60-69,1989

[7] E. Moshchuk, T. Bragin, S. D. Gribble and H. M. Levy, *A crawlerbased study of spyware on the Web*, (2006).

[8] Francisco Herrera, Genetic Fuzzy systems: A state of Art and new trends.

[9] Genetic Fuzzy Systems: Status, Critical Considerations and Future Directions by Francisco Herrera on International Journal of Computational Intelligence Research. ISSN 0973-1873 Vol.1, No.1 (2005), pp. 59-67.

[10] Introduction to Fuzzy Systems, Neural Network and Genetic Algorithms by Hideyuki TAKAGI in *Intelligent Systems: Fuzzy Logic, Neural Network and Genetic Algorithms* Ch.1 pp.1-33 by D.Ruan, Kluwer Academic Publishers, September 1997.

[11] J. Ma, L. K. Saul, S. Savage and G. M. Voelker, *Beyond blacklists: learning to detect malicious web sites from suspicious URLs, Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining,* ACM, Paris, France, 2009.

[12] Jorge Casillas, Brian Carse and Larry Bull, *Fuzzy-XCS: A Michigan Genetic Fuzzy System*, IEEE Transactions on Fuzzy Systems, vol. 15, no. 4, 2007.

[13] Kowaiczyk, R. *On numerical and linguistic quanti_cation in linguistic approximation*, IEEE International Conference on Systems, Man, and Cybernetics, 326{331.}, 1999

[14] L. Bin, H. Jianjun, L. Fang, W. Dawei, D. Daxiang and L. Zhaohui, *Malicious Web Pages Detection Based on Abnormaingl Visibility Recognition*, *E-Business and Information System Security, 2009.* EBISS '09. International Conference on, 2009, pp. 1-5.

[15] L. Shih-Fen, H. Yung-Tsung, C. Chia-Mei, J. Bingchiang and L. Chi- Sung, *Malicious Webpage Detection by Semantics-Aware Reasoning, Intelligent Systems Design and Applications, 2008*. ISDA '08. Eighth International Conference on, 2008.

[16] L.A. Zadeh, Fuzzy Sets. Information and Control 8, 338{353}, 1965.

[17] N. Provos, P. Mavrommatis, M. Abu and R. F. Monrose, *All your iframes point to us*, Google Inc, 2008.

[18] O.Cordon, F. Gomide, F.Herrerea, F.Hoffman and L. Magdalena "Ten years of genetic fuzzy systems: Current Framework and new trends", *Fuzzy Sets Syst.*, vol. 141, no. 1, pp. 5-31, 2004

[19] O.Cordon, F.Herrerea, F.Hoffman and L. Magdalena, *Genetic Fuzzy Systems. Evolutionary tuning and learning of Fuzzy Knowledge bases,* ser. Advances in Fuzzy Systems – Applications and Theory Series. Singapore: World Scientific, 2001, vol. 19

[20] On Advantages of Scheduling using Genetic Fuzzy Systems by Carsten Franke, Joachim Lepping and Uwe Schwiegelshohn

[21] Ossi Nykanen, *An Approach to Logic Programming with Type-1 Fuzzy Models Using Prolog,* IADIS International Conference Applied Computing, 2006.

[22] P. Liu and X. Wang, *Identification of Malicious Web Pages by Inductive Learning, Proceedings of the International Conference on Web Information Systems and Mining*, Springer-Verlag, Shanghai, China, 2009.

[23] P. Niels, R. Moheeb Abu and M. Panayiotis, *Cybercrime 2.0: When the Cloud Turns Dark*, Queue, 7 (2009), pp. 46-47.

[24] S. Xiaoyan, W. Yang, R. Jie, Z. Yuefei and L. Shengli, *Collecting Internet Malware Based on Client-side Honeypot, Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, 2008, pp. 1493-1498.

[25] Spears, W. M. & Anand, V., Charlotte, NC: Springer-Verlag, *A study of crossover operators in genetic programming*. Proceedings of the Sixth International Symposium on Methodologies for Intelligent Systems, 409-418, 1991.

[26] Syswerda, G., Vail, CO: Morgan Kaufmann, *Simulated crossover in genetic algorithms*. Proceedings of the Foundations of Genetic Algorithms Workshop, 1992

[27] T.P. Martin, J.F. Baldwin, B.W. Pilsworth, *The Implementation of Fprolog – A Fuzzy Prolog Interpreter*. Fuzzy Sets and Systems 23, 119 {129}, 1985.

[28] Technical Report on Ten Lecturers on Genetic Fuzzy Systems by Ulrich Bodenhofer, Francisco Herrera. Revised version of lecturer notes from “Preprints of the International Summer School: Advanced Control-Fuzzy, Neural, Genetic”, R.Mesiar, Ed.Slovak technical University Bratislava 1997. Pp. 1-69, ISBN.

[29] Time Complexity Ananlysis of Genetic – Fuzzy System for disease diagnosis by Ephzibah E.P. in *Advanced Computing: An International Journal) ACIJ), Vol.2, No.4*, July 2011.

[30] Toshinori Munakata, *Notes on implementing Fuzzy sets in Prolog,* Fuzzy Sets and System, 1998.

[31] V.Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer- Verlag, 1995.

[32] Van Lam Le, Ian Welch, Xiaoying Gao, Peter Komisarczuk, *Two-stage classification model to detect malicious web page,* International Conference on Advanced Information Networking and Applications,2011.

[33] Y.-M. Wang, D. Beck, X. Jiang and R. Roussev, *Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites that Exploit Browser Vulnerabilities*, IN NDSS (2006).

[34] Y.-T. Hou, Y. Chang, T. Chen, C.-S. Laih and C.-M. Chen, *Malicious web content detection by machine learning*, Expert Systems with Applications, In Press, Corrected Proof (2009).