Adaptive Fuzzy Classification-Rule Algorithm in Detection Malicious Web Sites from Suspicious URLS

Waleed Ead, Waeil Abdelwahed and Hatem Abdul-Kader
Faculty of Computers and Information Menoufiya University, Egypt

Abstract: *The Web has become a platform for supporting a wide range of criminal enterprises such as spam-advertised commerce, financial fraud. Although the precise commercial motivations behind these schemes may differ, the common thread among them is the requirement that unsuspecting users visit their sites. These visits can be driven by email, Web search results or links from other Web pages, but all require the user to take some action, such as clicking, that specifies the desired Uniform Resource Locator (URL). Malicious Web sites are a cornerstone of Internet criminal activities. As a result, there has been broad interest in developing systems to prevent the end user from visiting such sites. In this paper we propose Classification-rule discovery algorithm integrating artificial immune systems and fuzzy systems. The classification of new examples (antigens) considers not only the fitness of a fuzzy rule based on the entire training set, but also the affinity between the rule and the new example in training data set. This affinity must be greater than a threshold in order for the fuzzy rule to be activated. The proposed algorithm is considered to be an adaptive procedure for computing this threshold for each rule. Results are analyzed with respect to both predictive accuracy and rule set simplicity (comprehensibility). It is compared with C4.5 rules, which is a very popular data mining classification algorithm.*

Keywords: *Malicious Web sites, Prediction, classification-rule, artificial immune system, fuzzy system, security and protection.*

Received April 6, 2011; Accepted November 4, 2011

1. Introduction

If one could inform users beforehand that a particular URL was dangerous to visit, much of these problems could be reduced. To this end, the security community has responded by developing black-listing services, encapsulated in toolbars, appliances and search engines, which provide precisely this feedback. These blacklists are in turn constructed by a range of techniques including manual reporting, honeypots, and Web crawlers combined with site analysis heuristics. Many malicious sites are not blacklisted, either because they are too new, were never evaluated, or were evaluated incorrectly because it is masked [18]. For example, a malicious server may send benign versions of a page to honeypot IP addresses that belong to security practitioners, but send malicious versions to other clients [6]. To address this problem, some client-side systems analyze the content or behavior of a Web site as it is visited. But, in addition to run-time overhead, these approaches can expose the user to the very browser-based attacks that we seek to avoid [22], [20]. As new communications technologies drive new opportunities for commerce, they surely create new opportunities for criminal actors as well. The WWW is no exception to this pattern, and today millions of rogue Web sites advance a wide variety of scams including marketing false goods such as pharmaceuticals or luxury watches, financial fraud and propagating malware. What all of these activities have in common is the use of the Uniform Resource Locator (URL) as a vector to bring Internet users into their influence. Thus, each time a user decides whether to click on an unfamiliar URL they must implicitly evaluate the associated risk. Is that URL safe to click on, or will it expose the user to potential exploitation? .Not surprisingly, this can be a difficult judgment for individual users to make. As a result, security researchers have developed various systems to protect users from their uninformed choices. By far the most common technique, deployed in browser toolbars, Web filtering appliances and search engines, is blacklisting. [22] Using this approach, a third-party service compiles the names of known bad Web sites (labeled by combinations of user feedback, Web crawling and heuristic analysis of site content) and distributes the list to its subscribers [3].

The objective of this paper is to introduce an adaptive classification-rule algorithm for malicious web sites detection from suspicious URLs. The proposed algorithm is compared with C4.5 rules with respect to accuracy and rule simplicity. It is also tended to produce simpler rule sets more easily interpreted by a human user [15].

The paper is organized as follow: section 2 describes the problem of classification of URLs followed by URL features. Section 3 surveys some of the related works. Section 4 describes briefly the artificial immune system and the clonal selection principle followed by the fuzzy systems. Section 5 describes our proposed algorithm. Section 6 shows the experimental results followed by last section of conclusions and future works.

2. Problem Description

In this section, we provide a detailed discussion of our problem to classifying site reputation. We begin with an overview of the classification problem, followed by a discussion of the features we extract.

For our purposes, we treat URL reputation as a binary classification problem where positive examples are malicious URLs and negative examples are benign URLs. Classification problem can succeed if the distribution of feature values for malicious examples is different from benign examples, the training set shares the same feature distribution as the testing set, and the ground truth labels for the URLs are correct. Significantly, we classify sites based only on the relationship between URLs and the lexical and host-based features that characterize them, and we do not consider two other kinds of potentially useful sources of information for features: the URL's page content, and the context of the URL (e.g., the page or email in which the URL is embedded). Although this information has the potential to improve classification accuracy, we exclude it for a variety of reasons [18]. First, avoiding downloading page content is strictly safer for users. Second, classifying a URL with a trained model is a lightweight operation compared to first downloading the page and then using its contents for classification. Third, focusing on URL features makes the classifier applicable to any context in which URLs are found (Web pages, email, chat, calendars, games, etc.), rather than dependent on a particular application setting. Finally, reliably obtaining the malicious version of a page for both training and testing can become a difficult practical issue. Malicious sites have demonstrated the ability to "cloak" the content of their Web pages, i.e., serving different content to different clients [6]. For example, a malicious server may send benign versions of a page to honeypot IP addresses that belong to security practitioners, but send malicious versions to other clients.

2.1. URL Features

We categorize the features that we gather for URLs as being either lexical or host-based [18] as shown in figure 1.

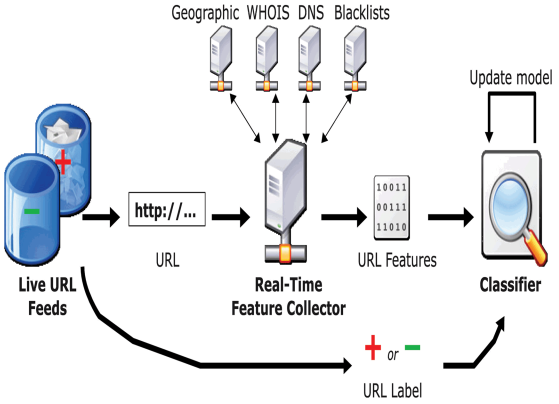

Figure1. Overview of real-time URL feed, feature collection, and classification infrastructure.

2.1.1. Lexical Features

The justification for using lexical features is that URLs to malicious sites tend to "look different" in the eyes of the users who see them. Hence, including lexical features allows us to methodically capture this property for classification purposes, and perhaps infer patterns in malicious URLs that we would otherwise miss through ad-hoc inspection. For the purpose of this discussion, we want to distinguish the two parts of a URL: the hostname and the path. As an example, with the URL www.geocities.com/usr/index.html, the hostname portion is www.geocities.com and the path portion is usr/index.html.

Lexical features are the textual properties of the URL itself (not the content of the page it references). We use a combination of features suggested by the studies of McGrath and Gupta [20] and Kolari et al. [13]. These properties include the length of the hostname, the length of the entire URL, as well as the number of dots in the URL, all of these are real-valued features. Additionally, we create a binary feature for each token in the hostname (delimited by '.') and in the path URL (strings delimited by '/', '?', '.', '=', '-' and '_'). This is also known as a "bag-of-words." Although we do not preserve the order of the tokens, we do make a distinction between tokens belonging to the hostname, the path, the top-level domain (TLD) and primary domain name (the domain name given to a registrar). More sophisticated techniques for modeling lexical features are available, such as Markov models of text. However, even with the bag-of-words representation, we can achieve very accurate classification results.

2.1.2. Host-Based Features

The reason for using host-based features is that malicious Web sites may be hosted in less reputable hosting centers, on machines that are not conventional web hosts, or through disreputable registrars. To an approximate degree, host based features can describe "where" malicious sites are hosted, "who" own them, and "how" they are managed.

The following are properties of the hosts (there could be multiple) that are identified by the hostname part of the URL:

- IP address properties: Is the IP address in a blacklist? Are the IPs of the A, MX or NS records in the same autonomous systems (ASes) or prefixes of one another? To what ASes or prefixes do they belong?
- WHOIS properties: What is the date of registration, update, and expiration? Who is the registrar? Who is the registrant? Is the WHOIS entry locked?
- Domain name properties*:* What is the time-to-live (TTL) value for the DNS records associated with the hostname?
- Geographic properties*:* To which continent/country/ city does the IP address belong? What is the speed of the uplink connection (broadband, dial-up, etc)?

3. Related Work

This section surveys some of related approaches in the general classification and URL classification.

The first AIS specifically designed for the classification task is AIRS [4]. In addition, it has been suggested that an AIS based on the Clonal selection principle, called CLONALG, can be used for classification in the context of pattern recognition [8], although originally proposed for other tasks. However, unlike the AIS algorithm proposed in this paper, neither AIRS nor CLONALG discovers comprehensible IF-THEN rules. Hence, neither of those two algorithms addresses the data mining goal of discovering comprehensible, interpretable knowledge. The AIS for discovering IF-THEN rules proposed in [16], is based on extending the negative selection algorithm with a genetic algorithm. We have avoided the use of the negative selection algorithm because this kind of AIS method has some conceptual problems in the context of the classification task, as discussed in [16].The work by Justin Ma et al. is the most closely related to our study [18]. They perform a comparative analysis by using support vector machine, Naive Bayes and logistic regression. Garera et al [17]. They use logistic regression over 18 hand-selected features to classify phishing URLs. Provos et al. [23]. Perform a study of drive-by exploit URLs and use a patented machine learning algorithm as a pre-filter for VM-based analysis.

4. Artificial Immune System and Fuzzy System

In this section we briefly describe the immune system followed by the fuzzy system.

4.1. Artificial Immune System

Computing and engineering have been enriched by the introduction of the biological ideas to help developing solutions for various problems. This can be exemplified by the artificial neural networks (ANN), evolutionary algorithms (EA), artificial life (ALife), and cellular automata (CA). There exist three different approaches, the first is: biologically motivated computing, under this umbrella the EA, ANN and artificial immune system (AIS), the second is computationally motivated biology, where computing provides models and inspiration for biology (i.e. ALife and CA). The third approach is computing with biological mechanisms, which involves the use of information processing capabilities of biological systems to replace or supplement the current silicon-based computers (e.g. Quantum and DNA computing) [11].

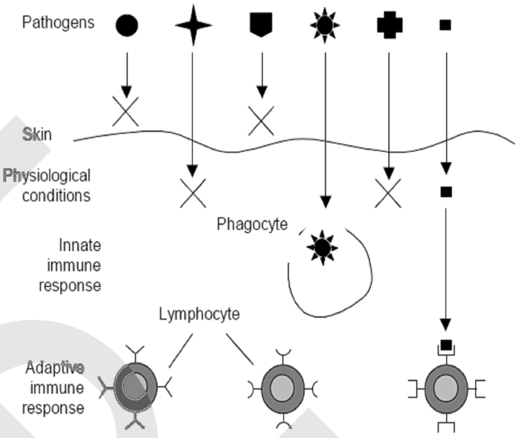

Figure 2. structure of the immune system.

The natural immune system [1] as depicted in figure 2 is based around a set of immune cells called *lymphocytes* comprised of *B and T-cells*. It is the manipulation of populations of these by various processes which give the system its dynamic nature. On the surface of each lymphocyte is a *receptor* and the binding of this receptor by chemical interactions to patterns presented on *antigens* which may activate this immune cell. A subset of the antigens is the *pathogens*, which are biological agents capable of harming the host (e.g. bacteria). Lymphocytes are created in the bone marrow and the shape of the receptor is determined by the use of gene libraries. These are libraries of genetic information, parts of which are concatenated with others in a semi-random fashion to code for a receptor shape almost unique to each lymphocyte. The main role of a lymphocyte in AIS is encoding and storing a point in the solution space or *shape space* [12]. The match between a receptor and an antigen may not be exact and so when a binding takes place it does so with strength called an *affinity*. If this affinity is high, the antigen is said to be within the lymphocyte's *recognition region*. As a lymphocyte may become activated by any antigen within this region a single lymphocyte may match a number of antigenic patterns, an important element of the noise tolerant nature of the immune system. When this binding takes place it stimulates an immune response from the lymphocyte and the cell begins to *clone* and *mutate*. The cloning takes place with a rate proportional to affinity and mutation with a rate inversely proportional to affinity in a process called *Clonal selection.* During this process strong selective pressures seek to maximize affinity with the antigen, thus increasing the efficiency of the response. Clonal selection constitutes the core of the immune system's adaptation mechanisms. This, however, is not the whole story as a T-cell requires two signals to become activated. Signal one is a binding via its receptor to an antigenic pattern, the second signal is called *co-stimulation* and is given by an *antigen presenting cell* as a confirmation that the bound antigen really is pathogenic. Once the pathogen has been removed a small number of clones with high affinities to the pathogen will live on to provide memory of the event. The immunological details of this process are under discussion, but this simple explanation of immune memory is of use in the artificial domain.

4.1.1. An Overview of the Clonal Selection Principle

The Clonal selection principle, or theory, is the algorithm used by the immune system to describe the basic features of an immune response to an antigenic stimulus. Clonal selection establishes the idea that only cells that recognize the antigens will proliferate where the rest will not, as depicted in figure 3 [11]. The most triggered cells selected as memory cells for future pathogens attacks and the rest mature into antibody secreting cells called plasma cells [1, 12, 10].

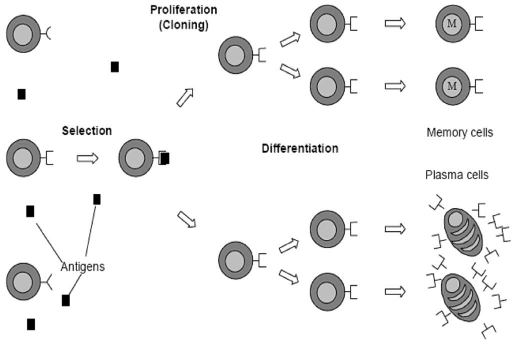

Figure 3. The Clonal selection principle.

4.2. Fuzzy Systems

Fuzzy systems use symbols, called linguistic terms, which have a well-defined semantics and are represented by membership functions of fuzzy sets. This allows the numerical processing of those symbols or concepts. Fuzzy systems are very effective in expressing the natural ambiguity and subjectivity of human reasoning [24-19].

Membership functions determine to which degree a given object belongs to a fuzzy set. In a fuzzy system this degree of membership varies from 0 to 1. Membership functions can take different forms, varying from the simplest ones (triangular functions) to more complex functions (parameterized by the user). According to [21], in a classification problem with *n* attributes, fuzzy rules can be written as: R_j : if x_1 is A_1^j and … and X_n is A_n^j then class C_j , j =1,...,N , where x=(x_1, ..., x_n) is an n-dimensional pattern vector, A_i^j (i=1,…,n) is the *i*-th attribute's linguistic value (e.g. small or large), C is the class predicted by the rule, and N is the number of fuzzy if-then rules. Hence, the antecedent ("IF part") of each fuzzy rule is specified by a combination of linguistic values.

5. Proposed Algorithm

The goal of the proposed algorithm is to discover a classification model (a rule set, in this work) that predicts the class of an example (a record) based on the values of predictor attributes for that example.

This work proposes a new algorithm for inducing a set of fuzzy classification rules based on an artificial immune system (AIS), a relatively new computational intelligence paradigm [9]. The proposed algorithm discovers a set of rules of the form "IF (fuzzy conditions) THEN (class)", whose interpretation is: if an example's attribute values satisfy the fuzzy conditions then the example belongs to the class predicted by the rule. The fuzzy representation of the rule conditions not only gives the system more flexibility to cope with uncertainties typically found in real-world applications, but also improves the comprehensibility of the rules [24, 19].

The proposed algorithm evolves a population of antibodies, where each antibody represents the antecedent (the "IF part") of a fuzzy classification rule. Each antigen represents an example (record, or case). More precisely, an antibody is encoded by a string with n genes, where n is the number of attributes. Each gene *i, i*=1,…,n, consists of two elements: (a) a value V_i specifying the value (or linguistic term) of the *i-th* attribute in the *i-th* rule condition; and (b) a boolean flag B_i indicating whether or not the *i-th* condition occurs in the classification rule decoded from the antibody. Hence, although all antibodies have the same genotype length, different antibodies represent rules with different number of conditions in their antecedent subject to the restriction that each decoded rule has at least one condition in its antecedent. This flexibility is essential in data mining, where the optimal number of conditions in each rule is unknown a priori.

The rule consequents (predicted classes) are not evolved by the AIS. Rather, all the antibodies of a given AIS run are associated with the same rule consequent, so that the algorithm is run multiple times to discover rules predicting different classes, as will be explained in more detail in the following subsection 5.1.

5.1. Discovering Rules from the Training Set

The proposed algorithm consists of: first, Main Algorithm (MA) as shown in figure 4. Second, the Learning Algorithm (LA) is based on clonal selection principle as shown in figure 5. Figure 6 show a schematic diagram of the proposed algorithm. It will be encapsulated in toolbars, appliances (e.g. Kaspersky internet security) and search engines to provide feedback of blacklists URLs.

```
Input: full training set
Output: fuzzy rules set
Rules set = 0
FOR EACH class value c in class values list DO
Values count = number of example of c in full training set
Training set = full training set
WHILE Values count > number of maximal uncovered examples
        Best rule = CLONAL−SELECTION−ALGORITHM (training set, c)
        Covered rule = COVER−SET (training set, best rule)
        Values count = Values count − size of covered
        ADD (rules set, best rule)
      END WHILE
END FOR EACH
Training set = full training set
FOR EACH best rule R in rules set DO
  MAXIMIZE−FITNESS (R, training set)
  COMPUTE−FITNESS (R, training set)
END FOR EACH
RETURN rules set
```

Figure 4.Pseudo-code of main algorithm.

```
Input: current TrainSet;
Output: the best evolved rule;
Create initial population of antibodies at random;
Compute fitness of each antibody;
FOR i = 1 to Number of Generations
            Perform tournament selection T times, getting T winners to be cloned;
   FOR EACH antibody to be cloned
            Produce C clones of the antibody, where C is proportional to fitness;
            FOR EACH just-produced clone
              Mutate clone with a rate inversely proportional to its fitness;
              Compute fitness of the clone;
     END FOR EACH clone;
   END FOR EACH antibody;
   Replace the T worst-fitness antibodies in the population by the T best-fitness clones;
END FOR i;
Return the rule whose antecedent consists of the antibody with the best fitness among all antibodies produced in all generations, and whose consequent consists of class c;
```

Figure 5. Pseudo-code of learning algorithm.

The MA (Main Algorithm) procedure starts by initializing the *DiscoveredRuleSet* to the empty set, and then it performs a loop over the classes to be predicted [6]. For each class, the algorithm initializes the *TrainSet* with the set of all examples in the training set and iteratively calls the LA (Learning algorithm) procedure, passing as parameters the current *TrainSet* and the class c to be predicted by all the candidate rules in the current run of that procedure. The LA procedure returns the best evolved rule, which is then stored in the variable *BestRule*. Next the algorithm adds the *BestRule* to the DiscoveredRuleSet and it removes from the current TrainSet the examples that have been correctly covered by the best-evolved rule. An example is correctly covered by a rule if and only if the example satisfies the rule antecedent and the example has the same class as predicted by the rule. In order to compute whether or not an example satisfies a rule antecedent we compute the affinity between the rule and the example, as follows.

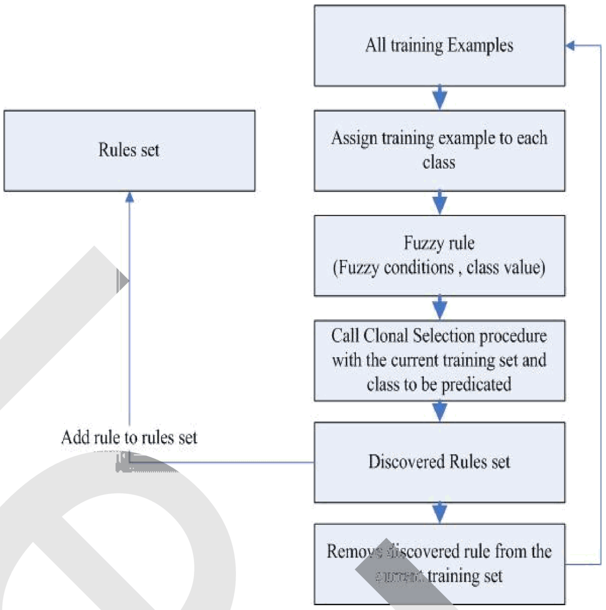
All training Examples
Rules set
Assign training example to each class
Fuzzy rule
(Fuzzy conditions , class value)
Call Clonal Selection procedure with the current training set and class to be predicated
Add rule to rules set
Discovered Rules set
Remove discovered rule from the current training set

Figure 6. schematic diagram of the proposed algorithm implementation.

First, for each condition in the rule decoded from an antibody, the algorithm computes the degree to which the original continuous value of the corresponding attribute (in the database) belongs to the fuzzy set associated with the rule condition. These degrees of membership are denoted by $\mu_{A1}(x_1)\ldots\ \mu_{An}(x_n)$ where n is the number of conditions in the rule. The next step is to compute the degree to which the example satisfies the rule antecedent as a whole. This is computed by applying the standard aggregation operator min to the $\mu_{A1}(x_1)$,…, $\mu_{An}(x_n)$ values. More precisely, the affinity between an antibody *j* and an antigen *k* is given by Equation (1):

$$Afin(k,j)=f\ (\mu_{A1}(x_1)\ ,\ \ldots\ ,\ \mu_{An}(x_n)) = \mu_{A1}(x_1)\ \wedge\ \ldots\ \wedge\ \mu_{An}(x_n) \qquad (1)$$

An example satisfies a rule (i.e., a rule is activated for that example) if the degree of affinity between the rule and the example is greater than an activation threshold, i.e., if $Afin(k,j) > L_j$, where L_j denotes the activation threshold for the *j*-th rule.

The WHILE loop is iteratively performed until the number of uncovered examples is smaller than a user-defined threshold *MaxUncovExamp*, so that this procedure discovers as many rules as necessary to cover the vast majority of the training examples. Finally, in the last step of the MA procedure we recompute the fitness of each rule in the DiscoveredRuleSet, by using the full training set.

The LA procedure starts by randomly creating an initial population of antibodies, where each antibody represents the antecedent of a fuzzy classification rule. For each rule, the system prunes the rule and computes the fitness of the antibody. Rule pruning has a twofold motivation: reducing the overfitting of rules to the data and improving the simplicity (comprehensibility) of the rules. The basic idea of this rule pruning procedure is that, the lower the predictive power of a condition, the more likely the condition will be removed from the rule. The predictive power of a condition is estimated by computing its information gain, a very popular heuristic measure of predictive power in data mining [14]. After rule pruning, the algorithm computes the fitness of each antibody, and then it performs the outer FOR loop over a fixed number of generations. The outer FOR loop starts by performing T tournament selection (with tournament size of 10) procedures, in order to select T winner antibodies that will be cloned in the next step. Tournament selection is well known and often used in evolutionary algorithms [2].

We now turn to the fitness function used by the LA procedure. The fitness of an antibody *Ab*, denoted by *fit(Ab),* is given by Equation (2):

$$fit(Ab) = \frac{TP}{TP + FN} x \frac{TN}{TN + FP} \qquad (2)$$

Where the variables TP, FN, TN and FP have the following meaning:

- TP = number of true positives, i.e. number of examples satisfying the rule and having the same class as predicted by the rule;
- FN = number of false negatives, i.e. number of examples that do not satisfy the rule but have the class predicted by the rule;
- TN = number of true negatives, i.e. number of examples that do not satisfy the rule and do not have the class predicted by the rule;
- FP = number of false positives, i.e. number of examples that satisfy the rule but do not have the class predicted by the rule.

This fitness function was proposed by [7] and has also been used by other evolutionary algorithms for discovering classification rules. However, in most projects using this function the discovered rules are crisp, whereas in our project the rules are fuzzy. Hence, in this project the computation of the TP, FN, TN and FP involves, for each example, measuring the degree of affinity (fuzzy matching) between the example and the rule. Note that the same affinity function (Equation (1)) and the same procedure for determining whether or not an example satisfies a rule are used in both the MA and the LA procedures.

The rules discovered from the training set are used to classify new examples in the test set (unseen during training) as follows. For each test example, the system identifies the rule(s) activated for that example. Recall that a rule *j* is activated for example k if the affinity between *j* and *k* is greater than the affinity threshold for rule *j*.

When classifying a test example, there are three possible cases. First, if all the rules activated for that example predict the same class, then the example is simply assigned to that class. Second, if there are two or more rules predicting different classes activated for that example, the system uses a conflict resolution strategy consisting of selecting the rule with the greatest value of the product of the affinity between the rule and the example (Equation (1)) by the fitness of the rule (Equation (2)), i.e., it chooses the class C given by Equation (3):

$$C=C_j=\max\,(Afin(k,j) \text{ x } fit(j)) \qquad (3)$$

Third, if there is no rule activated for the example, the example is classified by the “default rule”, which simply predicts the most frequent class in the training set [14].

6. Experimental Results

Basically, to evaluating the performance of any of supervised learning algorithms (Classification algorithms) is the idea of training and testing datasets. The training set contains examples of URLs from different classes (either malicious or benign) and is used to build the classification model. The testing set represents the unknown URLs examples that we wish to classify. As we know the class of each URL within the datasets we are able to evaluate the performance of the classifier by comparing the predicted class against the known class. To test and evaluate the algorithms we use k-fold cross validation. In this process the data set is divided into k subsets. Each time, one of the k subsets is used as the test set and the other k-1 subsets form the training set. Performance statistics are calculated across all k trials. This provides a good indication of how well the classifier will perform on unseen data. We use initially k=5, as you increase k values the more accuracy may be achieved. We compute following three standard measures:

- Accuracy: the percentage of correctly classified instances over the total number of instances.
- Precision: the number of class members classified correctly over the total number of instances classified as class members.

- Recall (or true positive rate): the number of class members classified correctly over the total number of class members.

We refer to the combination of accuracy, precision and recall using the term classification accuracy.

The proposed algorithm was evaluated in data sets are available from http://sysnet.ucsd.edu/projects/url. Table 1, shows the values of parameters in the proposed algorithm. Table 2, shows some of discovered rule set. The fitness of each rule must greater than the predefined threshold. Note that A1, A2,A3 … etc. represent the values of each feature attributes which classify the specified URL is benign or malicious.

Table 1. Parameters values to the proposed algorithm.

Parameter	Value	Description
NumAb	100	Number of antibody in the Initial Population
MinMutation	0.01	Likelihood maximum of mutation of each feature
ClonalRate	5	Number Maximal of clones that can be generate
PruningRate	0.5	Likelihood of an antibody will suffer edition of receptors
MaxUncovExamp	10	Number of uncovered examples into training set
MinCovRule	10	Minimal number of examples that a rule must cover

Table 2. Sample of discovered rule set.

```
Rule: 1
IF (A7!= " 0.165975") AND(A9 == " 0.72266") THEN "+1" FITNESS:
0.7510 THRESHOLD: 0.70
Rule: 2
IF (A4! = "I") AND (A7! = “0.836498") AND (A9 == “0.103448") THEN
"-1" FITNESS: 0.7537 THRESHOLD: 0.70
…
```

In figure 7, we feed our proposed algorithm with different datasets. The numbers of records in each dataset are in thousands. As the figure shows that the proposed algorithm achieves high accuracy rate than C4.5. We notice that the proposed algorithm achieve more accuracy than the other in large datasets.

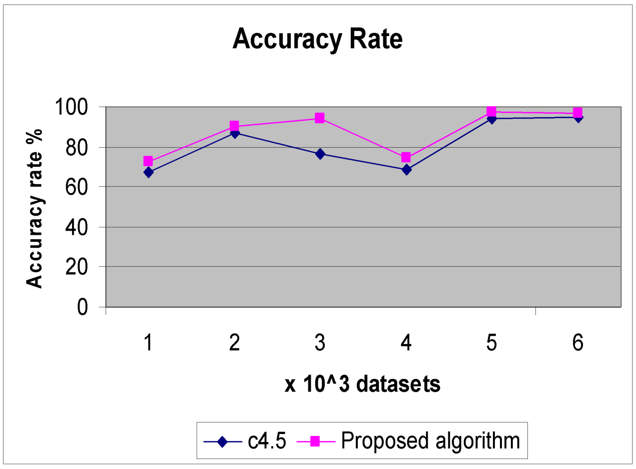

Figure 7. accuracy rate on different datasets.

There a comparison was carried out across criteria, namely the predictive accuracy of the discovered rule sets and their simplicity as discussed in the following. As previously mentioned, Predictive accuracy was measured by 5-fold cross validation procedure. In essence the data set in divided into 5 mutually exclusive partitions. Then the classification algorithm is run 5 times. Each time a different partition is used as the test set and the others is used as the training set. The results of the 5 runs (accuracy rate on the test set) are then averaged and reported in table 3 as the accuracy rate.

Table 3. accuracy rate on the test set.

Proposed algorithm	C4.5 Rules
94.66±1.70	87.32±2.79

The proposed algorithm as shown gives better results of accuracy rate of the discovered rule sets than C4.5.

We now turn to the results concerning the simplicity of the discovered rule set. This simplicity was measured as usual in the literature, by the number of the discovered rules and the total number of conditions in the antecedent of the discovered rules.

The results comparing the simplicity of the rule set discovered by the proposed algorithm and by C4.5 are reported in the table 4.

Table 4. Simplicity of the discovered rule set.

Average number of discovered rules		Average total number of conditions	
Proposed algorithm	C4.5 Rules	Proposed algorithm	C4.5 Rules
17.8 ± 0.20	33.8 ± 1.44	**22.4 ± 0.70**	63.4 ±1.53

As a result, the rules discovered by the proposed algorithm are not independent as the rules discovered by C4.5. This has the effect of reducing the simplicity of the rule discovered by the proposed algorithm by comparison with rules discovered by the C4.5.

As Table 4 shows the results of both proposed and C4.5 Rules with respect to the simplicity (comprehensibility) of the discovered rule set, measured by the average number of discovered rules and the average total number of conditions in all discovered rules. Knowledge discovered by the rules can be easily interpreted by the user. Hence, the user can validate discovered knowledge and combine it with her/his back-ground knowledge in order to make an intelligent decision, rather than blindly trusting the results of a black box. The accuracy of the proposed algorithm can be improved significantly by buffering more discovered rules in the clonal selection algorithm. In any case, this effect seems to be quite compensated by the fact that, overall, the size of the rule list discovered by the proposed algorithm is simpler the rule discovered by the C4.5, which is an important point in the context of the web mining.

7. Conclusion

Classification of such URLs from being suspicious or malicious based on their features will strong the idea of building the intelligent web server. This paper proposes classification-rule discovery algorithm which integrates artificial immune systems and fuzzy systems to increase the throughput of the algorithm. The proposed algorithm consists of two parts: first, a main algorithm (MA) procedure which sequentially covering all the training set to produce fuzzy rules. Second, a learning algorithm (LA) procedure for producing antibodies to new examples in the training set which considered new antigens. Each antibody (candidate solution) corresponds to a classification rule. The aim is to classify URLs automatically as either malicious or benign based on both lexical and host-based URL features. In this algorithm we tend to provide interpretable knowledge that is the goal of web mining. The classification of new examples (antigens) considers not only the fitness of a fuzzy rule based on the entire training set, but also the affinity between the rule and the new example. This affinity must be greater than a threshold in order for the fuzzy rule to be activated. It showed by the experiments that this proposed algorithms gains high accuracy rate especially in case of the complexity of the datasets. The average number of conditions and rules is significant less than other classification techniques (C4.5 rules). One of the major advantages of the proposed algorithm that it can handle millions of URLs whose features evolve over time, with high efficiency. A future work for more enhancing the performance of this work is adding fuzzy partition learning over the dataset by using clonal selection algorithm to automatic infer partitions for each attribute since population of antibodies represent a set of partitions, and antigen is a whole set of data. Another issue we can apply our algorithm to also classify email whether spam or not and the images especially in spam-advertised commerce.

References

[1] Andrew S, Alex A.F, and Jon T,"AISEC:an Artificial Immune System for E-mail Classification", Springer,2008.

[2] Back T., Fogel D.B., and Michalewicz, T., "Evolutionary Computation" Vol. 1. IoP Publishing, Oxford, UK, 2000.

[3] Bailey, M., Jahanian, F., Sinha S.," Shades of Grey: On the Effectiveness of Reputation-Based Blacklists", *Proceedings of the International Conference on Malicious and Unwanted Software (Malware),* Alexandria, Virginia, pp.57–64, 2008.

[4] Boggess L.C., Watkins A.B., "A Resource Limited Artificial Immune Classifier". *Proceedings* Con-gress on Evolutionary Computation. Pp.926-931, 2002.

[5] Boneh D., Chou N., Ledesma R., Mitchell J.C., Teraguchi Y., "Client-side defense against web based identity theft". In Proceedings of the Network and Distributed System Security Symposium (NDSS), 2004.

[6] Chen H., Niu,Y., Wang Y.M., *"A Quantitative Study of Forum Spamming Using Context-based Analysis".* In Proceedings of the Symposium on Network and Distributed System Security (NDSS), San Diego, CA, Mar 2007.

[7] Coutinho M.S., Lopes H.S., Lima W.C."An Evolutionary Approach to Simulate Cognitive Feedback Learning in Medical Domain". In: Sanchez, E., Shibata, T., Zadeh, L.A. (eds.), Genetic Algorithms and Fuzzy Logic Systems. World Scientific, Singapore, pp. 193-207,1997.

[8] Dasgupta D., Gonzales F.A., "An Immunogenic Technique to Detect Anomalies in Net-work Traffic". *Proceedings of Genetic and Evolutionary Computation. Morgan Kaufmann,* San Mateo.pp. 1081-1088,2002.

[9] Dasgupta D.," Artificial Immune Systems and Their Applications". Springer, Berlin, 1999.

[10] DeCastro L. N. and Zuben F. J., "Learning and optimization using the clonal selection principle", IEEE transactions on evolutionary computation", 2002.

[11] DeCastro L.N, Timmis J. (2002),*"Artificial Immune Systems: A New Computational Intelligence Approach*" Springer, 2002.

[12] DeCastro L.N., "Natural computing, Encyclopedia of information science and technology", vol. IV. Idea Group Inc, 2005.

[13] Finin T., Joshi A., Kolari P., "SVMs for the Blogosphere: Blog Identification and Splog Detection", In *Proceedings of the AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, Stanford, CA,2006.

[14] Frank, E., Witten I. H.,"Data Mining: Pratical Machine Learning Tools and Techniques with Java Implementation". Morgan Kaufmann, San Mateo, 2000.

[15] Freitas, A.A., Lopes H.S., Parpinelli R.S.," Data Mining With an Ant Algorithm Colony Optimization". IEEE T. Evol. Comput. 6:4 pp. 321-332, 2002.

[16] DeCastro L.N., "Natural computing, Encyclopedia of information science and technology", vol. IV. Idea Group Inc, 2005.

[17] Garera S.N., Provos M.C., and Rubin, A. D., "A Framework for Detection and Measurement of Phishing Attacks". *In Proceedings of the ACM Workshop on Rapid Malcode (WORM)*, Alexandria, VA, 2007.

[18] Geoffrey M., Justin Ma., Lawrence K. Saul, Stefan S, *"Beyond Blacklists: Learning to Detect*

Malicious Web Sites from Suspicious URLs". Proceedings of knowledge of data discovery 09, Paris, France, 2009.

[19] Gomide F., Pedrycz W., "An Introduction to Fuzzy Sets. Analysiss and Design". MIT Press, Cambridge, 1998.

[20] Gupta M., McGrath D.K "Behind Phishing: An Examination of Phisher Modi Operandi". In *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, San Francisco, CA,2008.

[21] Ishibuchi H., Nakashima T.,"Effect of Rule Weights in Fuzzy Rule-based Classification Systems". IEEE T. Fuzzy System.pp. 506-515,2001.

[22] Justin Ma., Lawrence K., Saul S., Savage G , and Voelker M.," Identifying Suspicious URLs: An Application of Large-Scale Online Learning". Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada, 2009.

[23] Monrose F., Mavrommatis P. M., Provos N, , Rajab A.," All Your iFRAMEs Point to Us". In *Proceedings of the USENIX Security Symposium*, San Jose, CA, 2008.

[24] Zadeh, L.A.,"Fuzzy Sets. Inform. Control",pp. 338-352,1965.

Hatem Abdul-kader obtained his B.S. and M.SC. (by research) both in Electrical Engineering from the Alexandria University , Faculty of Engineering , Egypt in 1990 and 1995 respectively. He obtained his Ph.D. degree in Electrical Engineering also from Alexandria University, Faculty of Engineering, Egypt in 2001 specializing in neural networks and applications. He is currently an Associate professor in Information systems department, Faculty of Computers and Information, Information systems department, Faculty of Computers and Information, Menoufiya University, Egypt since 2004. He has worked on a number of research topics and consulted for a number of organizations. He has contributed more than 30+ technical papers in the areas of neural networks, Database applications, Information security and Internet applications.

Waleed Mahmoud Ead was born on august 11, 1982 in Menouf, Menoufiya, Egypt. He received the B.S. from Faculty of Computers & Informatics, Zagazig University, Egypt in 2004 with grade very good with honor. Waleed is working as IT consultant for different IT systems. His M.SC. in information systems from faculty of computers and information, menufia university, Egypt in 2012. He is working in Higher Institute of Computer Science and Information Systems at October 6 university, as teaching assistance. Waleed has contributed more than 5+ technical papers in the areas of Web Usage Mining (WUM) and Artificial Immune system (AIS) in international conferences.