Network Intrusion Detection using Danger Theory and Genetic Algorithms

João Lima Santanelli and Fernando Buarque de Lima Neto

Computing Engineering Programme - Polythecnic School of Pernambuco
University of Pernambuco (UPE) - Recife, PE - Brazil
{jls2,fbln}@ecomp.poli.br

Abstract. One of the most concerning problems faced by practitioners, within the development and operation of IT communication networks, is the crescent number of network intrusion attempts. That kind of attacks compromise the integrity of several services provided through the Internet. This paper presents a technique capable of optimize Danger Theory-based Intrusion Detection Systems through the use of a Genetic Algorithms. To validate the approach, tests were performed on the KDD Cup 1999 database, provided by the University of California Irvine (UCI).

Keywords: intrusion detection, artificial immune system, danger theory, dendritic cells algorithm, genetic algorithms, optimization

1 Introduction

With the advancement of technology, society is becoming increasingly dependent on digital communications. Among these, Internet is doubtless the one gaining more ground over the time and thus, deserves also progressively more attention.

The exponentially growing of the amount of messaging throughout the computer networks is the perfect scenario for the rise of new threats. The flaws on the security systems due to Denial of Service Attacks, for example, is one of the problems that brings proportionally great economic loss to many countries around the world [1].

Computer network attacks are, mostly, not a goal, but a means for attaining harmful actions and crimes over the internet. The most evident prove of that are the websites security, which are never so secure that likes cannot be broken. However, along the time, there were other new motivations for attacks such as political questions, extortion and even contests for the supremacy among the attackers. Therefore, it is necessary to seek constantly improvement on network incident detection techniques. A system capable to detect malicious traffic in a computer network is called an Intrusion Detection System (IDS).

Among several techniques to attain cyber-attacks, Computational Intelligence (CI) stands out because of its autonomy, adaptability and lightweight. Within CI, Artificial Immune Systems (AIS) [2] seem to be a serious contestant because of its speed and flexibility in light of the scarce number of labeled examples. To explain the operation of the Human Immune System (HIS), Polly

Matzinger elaborated the Danger Theory [3], [4], [5], in 1994. The core idea is that organisms identify dangerous situations based on alarms generated by the affected cells or tissues. Drawing inspiration in the human defense mechanisms, it is possible to associate cyber threats to pathogenic agents and the network to the human body. Therefore, it is reasonable to create an IDS based on AIS.

Another technique that draws inspiration on Biology is the Genetic Algorithm (GA) [6], [7], [8]. They are, for example, able to optimize the solution search for a given complex combinatorial problem. The approach is comprised of associating the problem solutions to individuals of a population and, throughout genetic operations (such as selection, cross-over and mutation), allow them to evolve, originating more adapted individuals, which, then represent better candidate solutions for the problem at hand.

This paper intends to use Genetic Algorithms to optimize the parameters associated to the input patterns of a Danger Theory-based Intrusion Detection System (DT-IDS). To reach this goal, a system was implemented that consists of a Danger Theory-based detection algorithm, more precisely, the Dendritic Cell Algorithm (DCA) [9], and a Genetic Algorithm that optimizes the detection parameters of the first one. This system extracts the data from the KDD Cup 1999 database [10], provided by the University of California Irvine (UCI), and compares the results of the performed classification with the labels contained in database records.

This work is divided in six sections. The first one contextualizes the research, exposing the motivations and objectives of this work. The second one provides the theoretic background needed to understanding the problem and the solution and addresses basic concepts of Danger Theory, Genetic Algorithms and Cyber Attacks. The third one discusses related works, pointing out the most important aspects of each one and comparing their solutions with the one adopted in this paper. The fourth one describes the experimental sets assembled to validate the approach presented in this paper. The fifth one discusses the results obtained throughout the performed experiments. The sixth one presents the final considerations about this research and suggests some improvements that would be done in future works.

2 Background

2.1 Danger Theory

The Danger Theory was developed by Polly Matzinger [3], in 1994, in order to explain the operation of the Human Immune System. Previously, it was believed that the Human Immune System acts based on the "self-non-self" theory, recognizing elements that were part of the system or not.

The main idea of Matzinger's theory is that the immune responses to pathogenic agents are given not based on the "self-non-self" concept, but activated by danger signals - in a population dynamics manner.

Matzingers model includes three type of signals [4]:

1. *Signal Zero*: is the danger signal emitted by a compromised cell or tissue;
2. *Signal One*: indicates that an antigen has been detected by an immune cell; and
3. *Signal Two*: co-stimulatory signal emitted by an Antigen-Presenting Cell (APC) requesting an immune response from T-cells.

The Dendritic Cell Algorithm (DCA), proposed by J. Greensmith, is inspired by Matzingers model, that describes the Antigen Presentation Cell (APC) as a structure capable to activate virgin T cells, starting the immune response. The Dendritic Cell (DC) is a type of APC [3].

According to Greensmith [9], there are three types of Dendritic Cells:

1. *Immature DC*: collect parts of antigens and signals from other cells;
2. *Semi-mature DC*: identifies the local signals as "safe" and present the antigen to T cells, resulting in tolerance; and
3. *Mature DC*: identifies the local signals as "danger" and present the antigen to T cells, starting the immune reaction.

The DCA consists, basically, in generating a group of entities representing Dendritic Cells and exposing them to input patterns, that play the role of antigen parts. The pattern can be interpreted by each cell as "normal" or "anomaly". The classification is based on two values associated to the pattern: *p_safe* and *p_danger*, that, respectively, represent the probabilities of the pattern is associated with a safe or a dangerous situation [11]. These values are intrinsic to the input patterns, and, based on them, an immature DC can migrate to the other states: mature or semi-mature.

After the migration of the DCs, the result of the classification of the pattern as "safe" or "dangerous" can be performed using a voting mechanism based on the number of mature and semi-mature cells with the respective antigen associated to the pattern.

In a scenario with multidimensional input patterns, the challenge to be faced is how to calculate the *p_safe* and *p_danger* values associated with each input pattern. They could be represented as a weighted sum of the probabilities associated with each intrinsic attribute of the pattern. Certainly there will be an optimum set of weights that will make these values more accurate, resulting in a better classification process.

2.2 Genetic Algorithms

The first Genetic Algorithm was created by John H. Holland, in the early seventies [6]. They can be implemented in several manners, but the central idea consists of associating the solutions of a problem to individuals of a population (or chromosomes) and, throughout genetic operators, make them evolve, becoming more adapted. The function that measures the fitness of an individual of the population is called Fitness Function, and it is modeled according to the problem that is intended to be solved.

The genetic operators/concepts applied in this present study are (1) selection, (2) crossover, (3) mutation and (4) elitism [7].

1. *Selection*: To select the individuals that will survive to the next generation/that will be selected for reproduction; it was chosen the "roulette" mechanism, where the probability for an individual to be chosen is directly proportional to the value of its fitness function;
2. *Crossover*: For each pair of individuals chosen to reproduce, random attributes are selected to be permuted, originating two new individuals that may be added to the population poll;
3. *Mutation*: For each individual of the population, there is a probability of some of its attributes be randomly modified to originate a new individual; and
4. *Elitism*: The fittest individuals of each generation will pass automatically to the next, ensuring that the better solutions will be preserved.

In the previous section one can find methods to calculate *p_safe* and *p_danger* values for DCA, and there is an optimum set of weights capable to provide more accuracy to the classification process. To reach this goal, a good approach is the use of Genetic Algorithms.

In this present study, the set of weights that defines the values obtained for *p_safe* and *p_danger* is modeled as an individual of GA's population. More details about the operation of the algorithm will be provided in the fourth section of this paper, where is discussed the experimental methodology. For now, it suffices to say that these sets of weights are modified by the GA to provide more accurate results.

2.3 Cyber Attacks

Some intrusion experts believe that the most recent attack types are variants of known ones. Therefore, the known signature attacks must be sufficient to detect new types [10].

This work will use four frequent types of attacks, provided by the KDD Cup 1999 database: (1) DoS, (2) R2L, (3) U2R and (4) Probe.

1. *DoS*: Denial of service attack, which intend to turn the target unavailable;
2. *R2L*: Non-authorized remote access to a server;
3. *U2R*: Non-authorized access to super-user (root); and
4. *Probe*: Vulnerability exploiting of the target.

In the scenario of Intrusion Detection (as a classification kind of problem), there are four types of possible results: (1) true positive, (2) true negative, (3) false positive and (4) false negative.

1. *True Positive*: when the system notifies an intrusion that is being performed;
2. *True Negative*: when the system classifies correctly the normal traffic;
3. *False Positive*: When the system notifies an intrusion that does not exist in fact; and
4. *False Negative*: When the system does not notify an intrusion that is being performed.

The definitions above, make clear that the worst case among these four, considering a computer network intrusion scenario, is the *false negative*. This is because a *false negative* makes the system assume that there is no threat and, so, it is not necessary to take any counter-action, leaving the network unprotected. On the other hand, the *false positive* generates an alarm that leaves the network on alert. The only problem associated to this situation is that the network administrator is bothered, but it does not offer any harm to the network. So, in an Intrusion Detection System it is necessary that the *false negative* rate is as low as possible, and is is desirable that a low *false positive* rate is also achieved.

3 Related Works

Many works adopted KDD Cup 1999 Database [10] to test their theories. The use of Computational Intelligence to solve Intrusion Detection problems is very widespread.

Srinoy and Kurutach [12] combined Artificial Ant Clustering and K-Mean Particle Swarm Optimization (K-PSO) using the KDD Cup 1999 Database achieving a reduction of the dimensionality of the attribute set from 41 to 10.

The Particle Swarm Optimization (PSO) allied to Support Vector Machines (SVM) was used by Srinoy [13] to detect malicious network activities, reaching an accuracy of 96,11%, while the pure SVM, used by Somwang and Lilakiatsakun [14], performed 97,44% of correct classifications.

Dantziger and Lima Neto [2] purposed a hybrid approach using concepts of Danger Theory, Nave Bayes and Multi Agent Systems to perform intrusion detection in an IEEE 802.11 network.

In 2014, Zekri et al. [15] presented a summary of some major works on Artificial Immune System for Intrusion Detection Systems since 2005. Positive Selection, Generation of Detectors, Innate Immunity, Cooperative AIS and Clonal Selection figure among the techniques used in these works [15]. But it is notorious that the most recent studies focus on Dendritic Cell Algorithm (DCA) and Negative Selection Algorithm (NSA).

Zekri compared DCA with NSA in her work [15] and found most significant results for the first one, strengthening the idea that DCA is the AIS's state of art for IDS.

4 Experimental Methodology

This paper intends to present the use of Genetic Algorithms to optimize Danger Theory-based Intrusion Detection Systems (DT-IDS). For that we implemented a DT-IDS optimized with a Genetic Algorithm. The overview of our approach is shown in Fig. 1, which components are better explained in the following subsections.

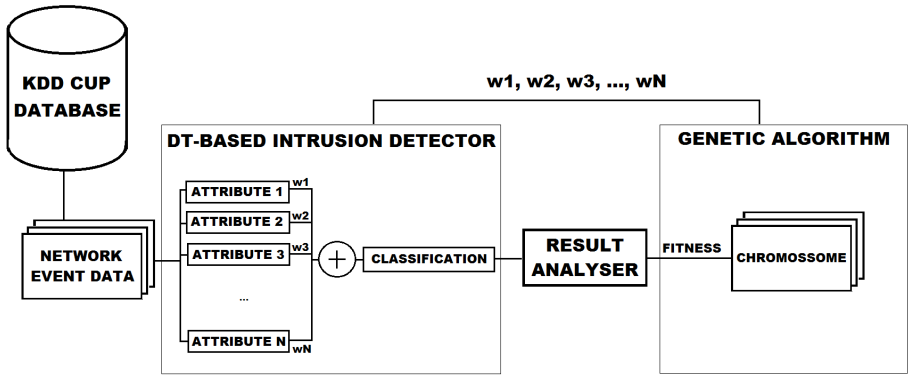

Fig. 1. Implemented model using Danger Theory and Genetic Algorithms

4.1 KDD Cup 1999 Database

The database for tests of our approach is provided by the KDD Cup 1999 [10] and is used for a contest where the competitors were challenged to create an Intrusion Detection System, classifying the network events as "good" or "bad" connections.

The amount of training and testing data, distributed by attack classes, are shown on the Table I.

Table 1. KDD Cup 1999 Database Figures

Event Type	**Amount of Data per Type**	
	Training	***Testing***
Normal	99.278	60.593
DoS	391.458	223.298
U2R	52	39
R2L	1.126	5.993
Probe	4.107	2.377
Unknown	-	18.729
TOTAL	**494.021**	**311.029**

The type of event defined as "unknown" represents the types of attacks that are not available in the training set, but as discussed in subsection 2.3, must be detected based on the known signature attacks.

4.2 Danger Theory-based Intrusion Detector

The implemented system has a decision core based on the Dendritic Cell Algorithm (DCA) [9]. The Dendritic Cells (DC) receive signal zero from the sensors

monitoring each attribute of a network event. These sensors could be associated to the cells and tissues compromised by the pathogenic agent. In each training iteration, the immature DCs are exposed to the antigen, represented by the set of attributes extracted from each network event registry obtained from the KDD Cup 1999 database, and so, they can evolve or not to the semi-mature or to the mature state. After some iterations, if the cell does not evolve, it is replaced by a new one. Each evolved cell migrates to a new group and is replaced by an immature cell, maintaining the immature DCs population size.

After training, the antigens represented by the testing registries are presented to the migrated cells and, throughout a voting mechanism, it is decided if the recognized pattern represents a threat or not.

Each network event registry extracted from the database represents one iteration of DCA. It means that DCAs training phase will have 494.021 iterations while the test phase will have 311.029 iterations, according to Table I, totalizing 805.050 iterations per execution.

The challenge to be faced is how to model a set of attributes (many of them continuous) as an input pattern capable to be recognized by Dendritic Cells. To solve this problem, we scanned the training data set and, for each attribute, we set an interval that goes from the minimum to the maximum value, dividing them into a discrete number of sub intervals.

The hardest aspects to tackle are the number of sub intervals and the differences between two input patterns associated to different types of attacks. On the other hand, a large number of sub intervals represents a larger domain of input patterns to be recognized by DCs and, for example, two patterns associated to the same network event type would be classified differently. To determine the better number of sub intervals to be used, several experiments were performed running DCA for different configurations and, according to the results exposed on Table II, we choose to divide each interval in three parts.

Another important aspect to be considered is that the *p_safe* value is associated to the probability of the antigen pattern represent a non-dangerous situation [11], but, in most examples using the canonical form of DCA found in literature, this value is not numerically equal to this probability. To model this problem, we calculate the *p_safe* value by multiplying that probability with a constant value. Several experiments were performed and we can note in Table II that the values for the *p_safe* multiplier between 1.49 and 1.53 present the best results showing an accuracy variation less than 0.07%. So, we decided to set the *p_safe* multiplier as 1.5.

To avoid any bias in the final results, all the tests performed to determine the parameters presented on Table II used, as training set, 10 percent of the training data provided by UCI in the KDD Cup Database [10]. The validation is performed upon the complete training database and each accuracy value was calculated five times.

On this work, DCAs *p_danger* value was modeled as 1 *p_safe*. The cells maturation depends on the antigens CMS value, defined as *p_safe* + *p_danger* [11]. The CMS value represents the relevance of antigens pattern. The higher

Table 2. DCA results for each number of sub intervals

Sub intervals	**Accuracy for p_safe multiplier**							
	1.43	*1.45*	*1.47*	*1.49*	*1.51*	*1.53*	*1.55*	*1.57*
2	0.9933	0.9934	0.9937	0.9938	0.9935	0.9933	0.9930	0.9929
3	0.9298	0.9326	0.9409	0.9955	0.9950	0.9949	0.9947	0.9946
4	0.9202	0.9234	0.9363	0.9856	0.9857	0.9937	0.9936	0.9935
5	0.9114	0.9143	0.9321	0.9759	0.9769	0.9929	0.9927	0.9926

is antigens *p_safe*, the larger is the certainty about its pattern to be associated with a non-dangerous situation. Analogously, the higher is antigens *p_danger*, the larger is the certainty about its pattern to be associated with a dangerous situation. But in this case, the sum *p_safe* + *p_danger* is always equal to 1. Therefore, to adapt our approach to the idea that CMS represents the relevance of antigens pattern, we defined it as the absolute value of the difference between *p_safe* and *p_danger*.

To be modeled as an antigen, the input pattern represented by the array containing the number of the sub interval in which is located the value of each attribute must be associated to a value of *p_safe* and *p_danger*, as discussed in subsection 2.1. These values determine how the DCs will respond to the exposure to the antigen.

The implemented decision logic to calculate the *p_safe* of each antigen is regulated by a set of weights (w1, w2, w3, , wN), as shown previously in the Fig. 1. Each attribute has its relevance associated to the respective weight and, therefore, it is desired to find the optimum weight set resulting in an optimal performance regarding security.

Each evaluated parameter is associated to a network attribute. For each one, a value between 0 and 1 is calculated, which represents the odds that the attribute is related to malicious traffic. Each value is multiplied by the weight representing the parameters relevance, obtaining a partial result. The antigens *p_safe* is calculated as the sum of these partial results.

Assigning the same value to the weights associated with each attribute would mean that all attributes have the same significance regarding the detection of anomalies. However, this is not true and it is not what is necessary for attaining high levels of intrusion detection. r Thus, we decided to associate all DT-IDS decision core weights to chromosomes of a Genetic Algorithm so it can reach an optimal solution based on detection rates obtained through the Result Analyzer, as in Fig. 1.

4.3 Genetic Algorithm-based Parameter Optimization

To optimize the DT-IDS, we associated the parameters used to calculate antigens *p_safe* values to individuals in a Genetic Algorithms population.

Due to the fact that DCA demands a considerable amount of time to be executed (approximately three minutes in these experiments, with ten immature

cells in the population) the Genetic Algorithm was modeled with a small population (eigth individuals per iteration). To assure that the best individuals will survive to the next generation, it was applied with 25% elitism.

The individuals selection for reproduction (crossover) is based on a "roulette" mechanism where the probability to be chosen is directly proportional to the individuals fitness. The fitness is set as the accuracy of DCA in classification of the training events using the associated parameters.

The reproduction is performed as a crossover mechanism that results in two new individuals. Each new individuals attribute is equal to the same attribute in one of the parents.

The selection of the individuals which will pass to the next generation is based in the "roulette" mechanism described previously in this section. The best 25% of the population is automatically selected.

To mitigate the stagnation of the algorithm in a local maximum, we set the mutation rate as 10 percent.

Stop conditions for the GA were established as the stagnation of the best fitness value for 10 rounds or the execution of 60 iterations. If at least one of the conditions is reached, the algorithm stops.

5 Simulation Results

To validate the approach, several experiments were performed. The first experimental set considers only known attack types, representing a less dynamical environment, where new types of attacks are not a real threat.

The second one represents a dynamical environment where new types of cyber-attacks emerge as new threats. The great challenge is to detect malicious activities that did not figure on the training data set, proving the adaptability of the approach.

5.1 Considering only known attacks

The results obtained throughout the first experimental set are shown in Table III, where the labels "Total" and "FP" represent respectively the total accuracy and the false positive rate of the detection process.

Table 3. DT-IDS optimized by Genetic Algorithm (known attacks)

Detection Accuracy						
Normal Traffic	**Attacks**				**Total**	**FP**
	DoS	**U2R**	**R2L**	**Probe**		
89,21%	99,42%	28,21%	2,92%	89,44%	95,23%	2,24%

5.2 Considering unknown attacks

The results obtained throughout the second experimental set are shown in Table IV, where the labels "Total" and "FP" represent respectively the total accuracy and the false positive rate of the detection process. The symbol "?" represents the accuracy in detecting unknown types of attacks.

Table 4. DT-IDS optimized by Genetic Algorithm

Detection Accuracy							
Normal	**Attacks**					**Total**	**FP**
Traffic	**DoS**	**U2R**	**R2L**	**Probe**	**?**		
89,21%	99,42%	28,21%	2,92%	89,44%	29,14%	91,25%	2,10%

6 Conclusion and future works

6.1 Discussion

The use of Genetic Algorithms for the optimization of Danger Theory-based Intrusion Detection Systems, as discussed in this paper, produced very interesting practical results.

Even though Tables III and IV show that the approach has not proved effective in detecting attacks of the type U2R, R2L (corresponding, respectively, to only 0.01% and to 0,22% of the training set) and the unknown attacks. This shows that the technique adds strong dependency on training set, not proving very able to adapt to a scenario with new threats. Fortunately, for known attacks our approach has been proved quite satisfactory (95,23% of accuracy). The results, for known attacks shows that the use of Genetic Algorithms as optimizer of Danger Theory-based classifiers is quite efficient, especially in DoS attack detection and false negative reduction.

An important achievement of this study is the reduction of false positive rates. A 2,1% false positive rate makes the system more independent of human intervention, considering that each false positive must be analyzed by the network administrator. The obtained false negative rate is very acceptable too.

Considering the aspects presented in this study, it is concluded that the use of the approach based on Danger Theory and optimized by Genetic Algorithms is recommended for less dynamic environments, where the emergence of new threats do not occur very often.

6.2 Future Work

The use of Genetic Algorithms can bring some inconvenience. One of the negative aspects is the need for large computational effort and, somehow acts against the benefits of AIS (DT), such as speed and flexibility.

With the advent of distributed computing, this problem can be easily alleviated, allowing the implementation of Genetic Algorithms with larger populations and more iterations, as parallelism decreases the time for completion of each step.

Thus, as future work, it is suggested to apply the concepts of Distributed Systems to enhance the performance of the Genetic Algorithm, allowing it possibly to reach better results.

Acknowledgements

Thanks to Computer Engineering Program of School of Engineering of University of Pernambuco and the Brazilian Army which made this research possible.

Thanks to Moisés Danziger, whose work [2],[3] and inspiration were seminal for the present study.

References

1. L-F Pau, "Business and social evaluation of denial of service attacks in view of scaling economic counter-measures" In 2010 IEEE/IFIP Network Operations and Management Symposium Workshops, pp. 126-133, 2010.
2. Danziger, M. and Lima Neto, F. B., "A Hybrid Approach for IEEE 802.11 Intrusion Detection Based on AIS, MAS and Nave Bayes", In International Journal of Computer Information Systems and Industrial Management Applications, Vol 3 (2011) pp. 193-201, Pernambuco, Brazil, 2011.
3. P. Matzinger, "Tolerance, danger and the extended family", Annual Reviews in Immunology, 12, pp. 991-1045, 1994.
4. Matzinger, P., "The Danger Model: a renewed sense of self", Science, 296, pp. 301-305, 2002.
5. Matzinger, P., "The Danger Model in Its Historical Context", Scandinavian Journal of Immunology, Vol 54, pp. 49, July/August 2001.
6. J. H. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Ardor, Michigan, 1975.
7. Razali, N. M., "Genetic Algorithm Performance with Different Selection Strategies" In Solving TSP, Proceedings of the World Congress on Engineering 2011, Vol II, 2011, London, U.K.
8. Q. C. Meng, "Genetic Algorithms and Their Application", Jinan, Publishing Company of Shandong University, August, 1995.
9. Greensmith J, Aickelin U. and Cayzer S., "Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection", In Proceedings ICARJS-2005, 4th International Conference on Artificial Immune Systems, LNCS, Springer-Verlag, Banff, Canada, 2005, pp.153-167.
10. KDD Cup 1999 Database
http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
{Accessed in Jun of 2016}
11. Brownlee, J., "Clever Algorithms: Nature-Inspired Programming Recipes", LuLu, 2011.
12. Srinoy, S. and Kurutach, W., "Combination Artificial Ant Clustering and K-PSO Clustering Approach to Network Security Model", In International Conference on Hybrid Information Technology, 2006.

13. Srinoy, S., "Intrusion Detection Model Based on Particle Swarm Optimization and Support Vector Machine", In Computational Intelligence in Security and Defense Applications. CISDA. IEEE Symposium, pp. 186-192, 2007.
14. Somwang, P. and Lilakiatsakun, W., "Computer Network Security Based On Support Vector Machine Approach", In 2011 11th International Conference on Control, Automation and Systems Oct. 26-29, 2011 in KINTEX, Gyeonggi-do, Korea, pp. 155-160, 2011.
15. Meriem Zekri, Labiba Souici-Meslati. "Immunological Approach for Intrusion Detection", In Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées (ARIMA), 2014, Vol 17, pp.221-240.
16. U. Aickelin, P. Bentley, S. Cayzer, J. Kim, J. McLeod, "Danger Theory: The link between AIS and IDS", In Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS), pp. 147-155, 2005.
17. J. Greensmith, U. Aickelin, G. Tedesco, "Information fusion for anomaly detection with the dendritics cell algorithm", In International Journal of Information Fusion, 2007.
18. Bhavin Shah, L. J. and Bhushan H. Trivedi, "Reducing Features of KDD CUP 1999 Dataset for Anomaly Detection Using Back Propagation Neural Network" In 2015 Fifth International Conference on Advanced Computing & Communication Technologies, pp. 247-251, 2015.
19. M. Gen and R. Cheng, "Genetic Algorithms and Engineering Design". Wiley Series in Engineering Design and Automation, 1997.
20. Jian Xu, Yongjun Xue, "Talking about the Intrusion Detection Technology", Sci-Tech Information Development & Economy, 24th, 2007.
21. Yan Zhou, Yi Han, "The Introduction of Intrusion Detection Technology", Computer Knowledge and Technology, March, 2004.