Clustering Analysis of BGP Update Messages for Anomaly Detection

Fei Xue Na Tang S. Felix Wu V. Rao Vemuri S. J. Ben Yoo
University of California at Davis
CA 95616, USA

{fxue, natang, rvemuri, sbyoo}@ucdavis.edu, wu@cs.ucdavis.edu

ABSTRACT

Detecting anomalous BGP routing dynamics is crucial to improving the stability of the Internet. In this paper, we investigate two unsupervised clustering methods, artificial immune network (aiNet) and evolving fuzzy neural network (EFuNN), for anomaly detection in BGP update messages. Both methods can categorize a large volume of raw data into a small number of clusters without a priori knowledge of the systems. We apply them to examine the BGP data collected during a period around the SQL worm attack. Experimental results show the effectiveness of clustering analysis in characterizing the BGP routing behaviors; the generated anomaly clusters are coincident in time with the anomalous BGP routing dynamics during the worm attack. Furthermore, we demonstrate that clustering analysis can effectively identify certain abnormal BGP messages that are worthy of further investigation. These results indicate that clustering analyses can help network operators and researchers to filter out the trivial events and to focus on the most important BGP events. This is an important step to determine the root cause of anomalous BGP routing behaviors.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network-- *Operations-network monitoring*

General Terms

Algorithms, Management, Performance

Keywords

Clustering, BGP routing dynamics, anomaly detection, artificial immune networks (aiNet), evolving fuzzy neural networks (EFuNN).

SIGCOMM'06 Workshop on Mining Network Data (MineNet-06), September 11–15, 2006, Pisa, Italy.

1. INTRODUCTION

BGP, the Internet's Border Gateway Protocol, is an essential component that enables inter-domain routing. It is of great importance to understand the BGP routing dynamics, since they affect the stability, connectivity, and availability of the Internet. In recent years, the research community has focused on numerous BGP convergence and instability problems [1-4]. More recently, anomaly detection of BGP routing dynamics has gained growing attention, with several methods being proposed in literature [5, 6].

With the increased complexity of the Internet, the analysis of operational BGP dynamics is a challenging issue. It is often difficult to manually pinpoint BGP anomalies and their root causes in real time. Suitable techniques are needed to learn the patterns embedded in the BGP data and to discover the knowledge for anomaly detection. Machine learning and data mining techniques have great potential to meet this need, as evidenced by their successful applications in the field of intrusion detection [7-9]. These learning techniques can be roughly classified into two categories: unsupervised (clustering) and supervised (classification). The supervised learning techniques require the label information associated with the training data. However, there is no systematic approach in practice to consistently label a BGP event as normal or abnormal. Since the operations of BGP involve multiple administrative domains, in certain circumstances, it is impossible for network administrators to unambiguously label a BGP event based on the single-domain knowledge. Considering the dynamic nature of BGP routing and the difficulty of labeling BGP data, clustering or unsupervised learning is more appropriate for BGP anomaly detection than supervised-learning-based methods.

In this paper, we will conduct the unsupervised clustering analysis on BGP data and investigate its applications to BGP anomaly detection. The clustering methods used in this work include artificial immune network (aiNet) [10-12] and unsupervised evolving fuzzy neural network (EFuNN) [13, 14]. Both are biologically inspired learning algorithms capable of partitioning the data space. aiNet is an immune system

based algorithm to explore "self/no-self" space separation for clustering, while EFuNN uses a neuro-fuzzy structure to perform clustering in an online adaptive fashion through unsupervised learning. An important advantage of clustering methods is their abilities to naturally categorize a large volume of raw data into a small number of clusters without a priori knowledge of the underlying systems. In particular, the online adaptive clustering methods like EFuNN can dynamically generate a new cluster for patterns unseen before. In the context of anomaly detection, this implies that on-line clustering has the potential to find new anomalous behaviors through its adaptive learning capabilities. This paper will demonstrate the advantages of the clustering-based anomaly detection through a set of experiments on BGP update messages. Experimental results show that clustering analysis can effectively identify certain abnormal BGP messages that are worthy of further investigation. Since clustering analysis provides an abstraction from the unstructured data space to the representative clusters, it enables network operators and researchers to filter out the trivial events and to put their focuses on the most important clusters. We will present case studies to demonstrate how to utilize the clustering results for further examination such as root cause analysis. Our experimental results indicate that clustering analysis is the important first step for anomaly detection and root cause analysis of BGP routing dynamics.

This paper is organized as follows. Section 2 presents the key concepts of aiNet and EFuNN. Section 3 describes the dataset of BGP update messages and presents feature extraction procedures used in the experiments. Section 4 discusses the experimental results on the clustering analysis. This paper concludes with a summary in Section 5.

2. CLUSTERING METHODS

2.1 Artificial Immune Networks (aiNet)

Artificial immune systems (AIS), a new family of biologically-inspired learning algorithms, have been successfully applied to many application areas by exploring immune mechanisms [12, 15]. The aiNet is one such AIS approach for data clustering. We first summarize the immune principles involved in the aiNet, and then describe the aiNet algorithm.

In the presence of infections, the immune system randomly produces many B-cells, which secrete antibodies to bind antigens and finally destroy them. The affinity between an antigen and an antibody describes the strength of their binding. The B-cells with high affinity to antigens are cloned. These cloned cells can easily recognize and bind antigens, and are called memory cells. This cloning process is called *clonal selection*. *Memory cells* have a longer life than normal B-cells and are thus useful when a similar infection occurs in future. The B-cells that have low affinity are mutated to obtain comparatively higher affinity to the antigens. This process of increasing affinity is called *affinity maturation*. Another related immune principle is the *immune network theory*, which indicates that the immune system involves not only antibody-antigen interactions but also antibody-antibody interactions. Antibodies are connected to form a network representing an internal image of antigens. The general immune network model can be generalized into the following formula [16]:

$$\text{RPV} = \begin{matrix}\text{Influx of} \\ \text{new cells}\end{matrix} - \begin{matrix}\text{Death of} \\ \text{unstimulated} \\ \text{cells}\end{matrix} + \begin{matrix}\text{reproduction of} \\ \text{stimulated cells}\end{matrix},$$

where RPV stands for the rate of the population variant of the network.

In the aiNet algorithm [10], each data point is treated as an antigen (*Ag*). It evolves a population of antibodies (*Ab*) based on the *immune network theory*, *clonal selection* and *affinity maturation*. These antibodies form a network to represent the antigens. Eventually clusters are automatically generated via hierarchical agglomerative clustering (HAC). In detail, first we randomly generate a set of *Abs* and put them into an empty memory matrix called *M*, and then follow the steps below:

(1) Affinity calculation: Calculate the affinity between the current *Ag* and each *Ab* from *M*. **(2)** Clonal selection: Select a subset of *Abs* with the highest affinity and clone them. The clone size is proportional to the affinities of *Abs*. **(3)** Affinity maturation: Mutate each *Ab* toward the current *Ag* with a rate inversely proportional to its affinity. **(4)** Reselection: Calculate the affinity between each *Ab* and the current *Ag*; Reselect a subset of *Abs* with highest affinity and remove the *Abs* with low affinity to the current *Ag*. **(5)** Network suppression: Remove redundant *Abs* and insert the resulting *Abs* into *M*. **(6)** Repeat (1)-(5) for each *Ag*. The memory matrix *M* would eventually contain the *memory cells*, i.e., the *Abs* that bind the *Ag* closely for each *Ag*. **(7)** Suppress *M*: Remove redundant *Abs* in *M* to maintain an appropriate size. **(8)** Add a set of new randomly generated *Abs* into *M*. **(9)** Repeat (1)-(8) until a pre-defined number of iterations are reached.

After the antibodies are built, clusters are detected from these antibodies via HAC. The suppression threshold σ_s (step (5) and (7)), the threshold to eliminate redundant *Ab*s, is the most essential parameter. It controls final network size and is responsible for the network plasticity. This parameter is generally determined based on the data dimension.

2.2 Evolving Fuzzy Neural Networks (EFuNN) with Unsupervised Learning

EFuNN is one of the evolving connectionist systems developed by Kasabov [13, 14] that is capable of modeling evolving processes through incremental online learning. It can learn spatial temporal sequences in an adaptive way through one pass learning and automatically adapt its structure and parameter values during the operation of the system. The unsupervised EFuNN provides one-pass clustering of an input data stream, where there is no predefined number of clusters. It is an effective clustering approach capable of tackling the "concept drift" problem in practical applications, and has been successfully applied to bio-informatics, speech recognition, and intrusion detection [9, 13, 14].

EFuNN uses a three-layer neuro-fuzzy structure for clustering, which includes an input layer, a fuzzy input layer, and a pattern layer. Fig.1 shows this structure. The nodes at the input layer read the input vector $X = \{X_1, X_2, \ldots, X_n\}$ into the system. The fuzzy input layer transforms the original input vector into a fuzzy input vector X_f by using a membership function attached to the fuzzy input nodes. The pattern layer represents previously learned patterns, with each pattern node corresponding to a cluster in the input space. Each pattern node i uses a weight vector W_i to represent the corresponding cluster.

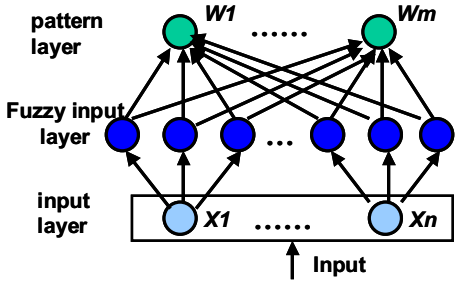


Fig.1. Three-layer structure of EFuNN.

EFuNN uses a metric called local normalized fuzzy distance to measure the distance between a fuzzy input vector X_f and a weight vector W_i associated with pattern node i. This metric is defined as $D(X_f, W_i) = \|X_f - W_i\| / \|X_f + W_i\|$, where $\|X - Y\|$ denotes the sum of all the absolute values of a vector that is obtained after vector subtraction (or summation in case of $\|X + Y\|$). Based on the distance between a new input vector and the existing patterns, EFuNN either assigns the new input vector to one of the existing patterns and updates the corresponding pattern weight vector, or creates a new pattern node for the new input. To perform clustering for each new input vector X, we use the following algorithm: (1) calculate the fuzzy input vector X_f; (2) evaluate the local normalized fuzzy distance D between X_f and the existing pattern weight vectors; (3) calculate the activation A of the pattern layer. For pattern node i, its activation $A(i)$ is defined as $A(i) = f(D(X_f, W_i))$, where f can be a simple linear function, e.g., $A(i) = 1 - D(X_f, W_i)$; (4) find the closest pattern node k to the fuzzy input vector X_f; (5) If $A(k) \geq S_k$ (S_k: sensitivity threshold for pattern node k), allocate the new input X to pattern node k (i.e., X is categorized into cluster k) and update the weight vector W_k according to the following vector operations: $W_k^{(new)} = W_k^{(old)} + l_k(X_f - W_k^{(old)})$, where l_k is the learning rate. Otherwise, create a new pattern node for the input X (in this case, X is categorized into this new cluster) and assign the weight vector of the new pattern node as X_f.

The parameters of sensitivity threshold S and learning rate l can be either static or self-adjustable. In our implementation, the learning rate l is pattern-specific so that the pattern node that has more instance members will change less when it accommodates a new instance. For sensitivity threshold S, all the pattern nodes share the same static value.

3. BGP UPDATE DATASET AND FEATURE EXTRACTION

In this work, we use the BGP dataset collected by the Routing Information Service of RIPE [17]. The dataset consists of the information about BGP update messages for a set of IP prefixes in different AS peers. We choose the BGP messages from two different AS peers (AS 1103 and AS 3549), which were observed during January 2003 for the IP prefix 166.111.0.0/16. The reason we choose these messages is that we attempt to correlate our clustering analysis with BGP routing dynamics during the SQL worm outbreak, which attacked Internet on January 25, 2003. While SQL worm does not intent to attack BGP routing architecture, BGP has been impacted during that period. Similar to previous work on statistics-based BGP anomaly detection [6], we extract three features for each BGP update message based on its arrival time and AS path information:

BGP updates message arrival frequency: this feature is related to the inter-arrival time of a BGP update message and characterizes the BGP update burst. When a new update arrives, we calculate this feature as $F_n = 1 + F_{n-1} \times 2^{-r \times \Delta t}$, where r is the decay factor and Δt is the inter-arrival time between the

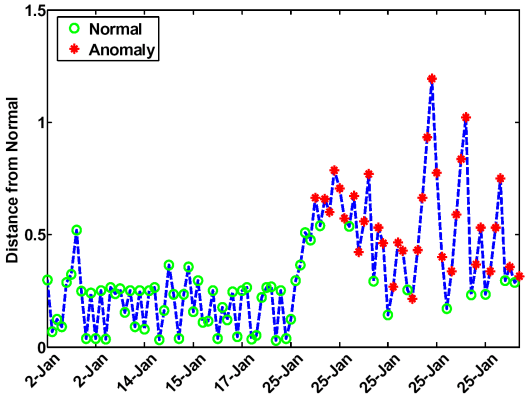
current and the previous update. In this work, the decay factor r is set to be 1/300.

Number of AS paths: this feature measures the variation of the number of AS paths and is calculated as $Q_n = N_{new\_aspaths} + Q_{n-1} \times 2^{-r \times \Delta t}$, where current Q is the sum of the number of new AS paths detected in the current update message and decayed previous Q.

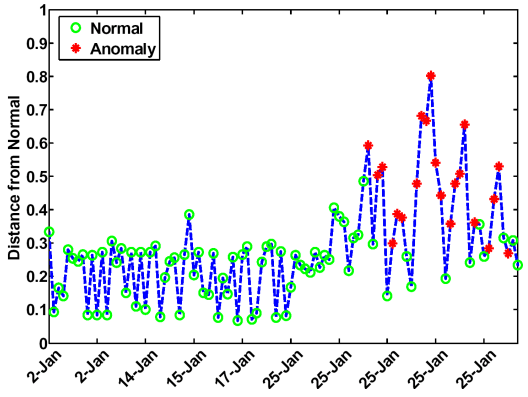
AS path occurrence frequency: this feature measures the frequency distribution of AS paths occurrence and is calculated as $P_n = \sum_{m=1}^{M} \left[(g_{m,n} - f_m)^2 / V_m \right]$, where f_m is the relative frequency with the m-th AS path has occurred in the history, $g_{m,n}$ is the relative frequency with which the m-th AS path has occurred in the recent past (which ends at the nth received update message), V_m is the approximate variance of the $g_{m,n}$. Detailed computation of these variables is discussed in [18].

4. EXPERIMENTAL RESULTS

We apply aiNet and EFuNN clustering methods on the BGP data to test whether they are able to partition data space of BGP messages into clusters and whether this clustering analysis can help analyze BGP routing dynamics.

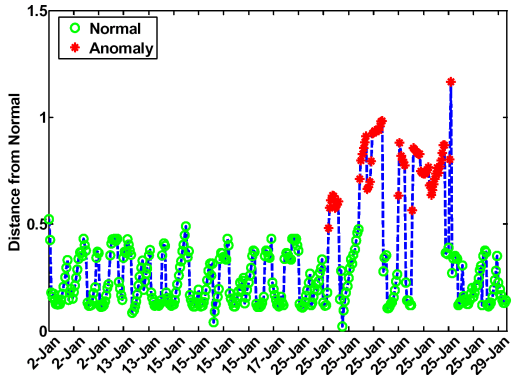


(a) clustering method: aiNet

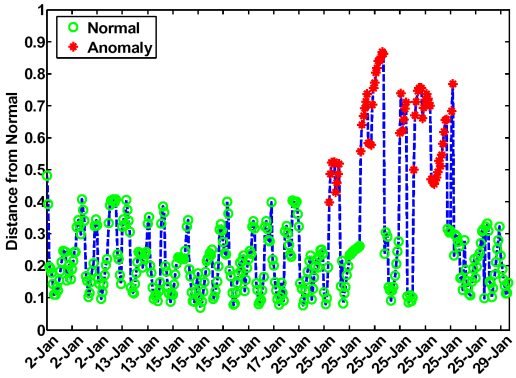


(b) clustering method: EFuNN

Fig.2 Clustering analysis for dataset #1 (AS 1103).



(a) clustering method: aiNet



(b) clustering method: EFuNN

Fig3. Clustering analysis for dataset #2(AS 3549).

Figs. 2 and 3 show the clustering results on two BGP datasets; dataset #1 is from AS1103 for the prefix 166.111.00/16, and dataset #2 is from AS3549 for the same prefix. Both aiNet and EFuNN have successfully clustered the BGP update messages into two categories: the majority of the BGP message before and after the worm attacks are clustered into the "normal cluster", while many BGP messages observed during the worm day (Jan. 25 2003) are in the "abnormal cluster". We will present the detailed discussion on these results in the next two paragraphs. These figures show the distance between the feature vector of each BGP message and the center of the "normal cluster" (Y-axis) with the date information of the BGP messages (X-axis). The green circles indicate the updates in the "normal cluster", while the red crossed data points indicate the "abnormal cluster" updates. As Figs. 2 and 3 show, the generated "anomaly clusters" from clustering analysis are coincident in time with the anomalous BGP routing dynamics observed during the worm attack.

For aiNet clustering, we set the suppression threshold σ_s to be 0.3 because of the relatively low dimension (3-D) of the BGP data. The aiNet first generates a small set of antibodies to represent the BGP updates (antigens) via an evolutionary process and then detects two clusters among the constructed

antibodies via HAC. For the dataset #1, aiNet detected the first anomalous attack at 05:49:08 GMT on the attack day (note that the SQL worm began propagating itself in the wild around 05:30 GMT [19]) and categorized 32 messages (out of total 48 messages observed in the worm day) as "anomalous". For the dataset #2, the "anomaly cluster" emerges at 06:16:51 GMT, and there are 63 messages (out of 161 messages observed in the worm day) in the "anomaly cluster".

For EFuNN clustering, we use the triangular membership function in the fuzzy input layer, and set the sensitivity threshold *S* to be 0.45 for dataset #1 and 0.6 for dataset #2. As an on-line adaptive clustering method, the EFuNN either updates the centers of existing clusters or creates a new cluster when a new input arrives. For both datasets, the EFuNN method generates only one cluster prior to the worm day, i.e., all BGP updates observed before the attack day are classified into the "normal cluster". For the dataset #1, the EFuNN method creates the "anomaly cluster" at 06:06 GMT on the attack day, and it categorizes 21 messages into the "anomaly cluster". For the dataset #2, the "anomaly cluster" emerges at 06:16, and there are 63 messages in the "anomaly cluster".

Figs. 2 and 3 also show that the BGP routing dynamics from two observation points (AS 1103 and AS 3549) exhibit similar patterns, which are generally consistent across time. For the data from different ASes, the generated anomaly clusters emerged around the same time. Exploring this spatio-temporal correlation among multiple ASes could provide helpful information to understand the global BGP operations, which is a topic deserving further investigation.

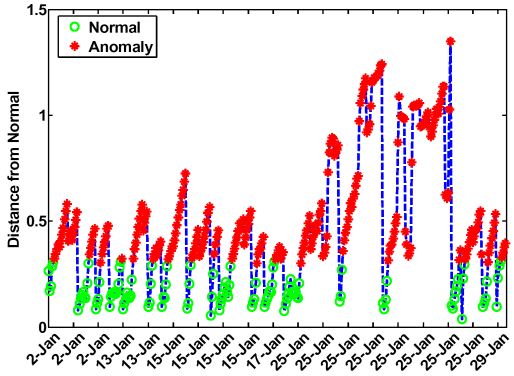


Fig. 4. Clustering analysis on dataset #2(AS3549) using K-means clustering method.

For comparison, we also use the simple K-means clustering method, which was mentioned in previous work [5] but there was no results on its performance. Fig.4 shows the analysis results on dataset from AS3549. For this dataset, the K-means clustering performs badly in partitioning BGP messages; there is no obvious correlation in time between the generated clusters and the BGP routing dynamics. It tends to classify lots of BGP update messages before the worm day into the "anomaly cluster", leading to a high false positive rate. In this sense, aiNet and EFuNN are more accurate than K-means methods, and achieve better clustering performance on BGP data.

As previous work [6] has noticed, it is difficult to validate the identified anomalies, because conducting validation requires the necessary information from a real BGP run-time environment and may involve cooperation among different ASes. The lack of such information makes it impossible to provide accurate analysis for speculative anomalies [6]. This is also the case for our study. While it is difficult to thoroughly evaluate our methods at the current stage, the generated "anomaly clusters" from aiNet and EFuNN analyses are coincident in time with the anomalous BGP routing dynamics during the worm attacks, indicating the effectiveness of these methods in partitioning BGP messages. Through naturally categorizing BGP data into a small number of clusters, clustering analysis can help operators and researchers to filter out the trivial events and to focus on the most important BGP events for further examination.

Table 1. Abnormal BGP messages from AS 1103.

Time	AS PATH
07:10:02	1103 11537 22388 7660 2516 3561 1239 9405 4538
07:12:21	1103 11537 22388 7660 2516 3561 1239 9407 9407 4538
07:13:44	Path withdrawal
07:15:11	1103 11537 9405 4538
07:19:31	1103 11537 22388 7660 4538

Based on the clustering results, we conduct further examination on some BGP message sequences in the "anomaly clusters", and attempt to determine their root causes. Table 1 shows a sequence of identified anomalous BGP updates from AS 1103 around 7:00 on Jan.25 2003. Recall that the AS path listed here is for the prefix 166.111.0.0/16. Prior to the worm day, this prefix had a steady path of (1103 11537 9405 4538). During the worm day, this previously stable path was seldom used, and the AS 1103's path to this prefix became unstable and changed frequently. At time 7:13:44, the BGP withdrawal message causes the prefix unreachable. By examining the AS-path changing patterns in these abnormal messages, we speculate that the peer link between AS 9405 and AS 4358 was instable during the worm outbreak, which could be the possible root cause of the BGP routing instability. In fact, both AS 9405 and AS 4538 are owned by a national research network in China, and the report in [20] confirmed that the connectivity of China's networks has been severely affected by the SQL worm. This example shows that our clustering

analysis method can effectively identify certain abnormal BGP messages that are worthy of further investigation. The detailed examination on these messages could provide an insightful understanding of the BGP operations in the underlying ASes.

5. CONCLUSION

In this paper, we investigated aiNet and EFuNN for clustering analysis of BGP data, and discussed their applications to detect anomalous BGP update messages. Experimental results indicated that both methods are effective in characterizing the BGP routing behaviors; the generated "anomaly clusters" are coincident in time with the anomalous BGP routing dynamics observed during the worm attacks. In addition, we observed that aiNet and EFuNN are more accurate than K-means clustering in partitioning BGP messages. Moreover, we demonstrated that aiNet and EFuNN clustering could effectively identify certain abnormal BGP messages that are worthy of detailed investigation. These abnormal BGP messages can be further examined to determine the source of anomalous behaviors. These results show that clustering analysis is an important starting point for anomaly detection and root cause analysis of BGP routing dynamics.

In the future work, we plan to apply the clustering approaches to examine BGP data measured in a controllable testbed environment, which can enable us to conduct in-depth evaluation of the clustering results.

6. ACKNOWLEDGMENT

This work is supported in part by NSF grant #0520333 and AFOSR grant FA 9550-04-1-0159. We would like to thank EYu Wang and Spencer Mathews for feature extraction of BGP data.

7. REFERENCES

[1] J. Cowie, et al., "Global Routing Instabilities Triggered by Code Red II and Nimda Worm Attacks," Rensys Corporation White Paper, Hanover, NH 2001.

[2] C. Labovitz, et al., "Internet Routing Instability," *IEEE/ACM transaction on Networking*, vol. 6, pp. 515-528, 1998.

[3] M. Lad, et al., "Analysis of BGP Update Surge during Slammer Worm Attack," in Proceeding of Fifth International Workshop on Distributed Computing, Calcutta, India, 2003.

[4] D. Pei, et al., "Improving BGP convergence through consistency assertions," IEEE INFOCOM, 2002.

[5] J. Zhang, et al., "Learning-based anomaly detection in BGP updates," SIGCOMM'05 workshops, 2005.

[6] K. Zhang, et al., "On detection of anomalous routing dynamics in BGP," in Proceedings of Third International IFIP-TC6 Networking Conference, Athens, Greece, May 2004.

[7] K. Julisch, "Data mining for intrusion detection: a critical review," in *Data Mining for Security Applications*: Kluwer, 2002.

[8] T. Lane and C. E. Brodley, "Temporal sequence learning and data reduction for anomaly detection," *ACM Transactions on information and system security*, vol. 2, pp. 295-331, 1999.

[9] Y. Liao, et al., "A General Framework for Adaptive Anomaly Detection with Evolving Connectionist Systems," in Proceedings of SIAM International Conference on Data Mining, Lake Buena Vista, FL, 2004.

[10] L. N. D. Castro and F. J. V. Zuben, "AiNet: an Artificial Immune Network for Data Analysis.," in *Data Mining: A Heuristic Approach*, H. A. Abbass, R. A. Sarker, and C. S. Newton, Eds.: Idea Group Publishing, 2001, pp. 231-259.

[11] L. N. D. Castro and F. J. V. Zuben, "AiNet: an Artificial Immune Network for Data Analysis. In Data Mining: A Heuristic Approach," H. A. Abbass, R. A. Sarker, and C. S. Newton, Eds.: Idea Group Publishing, 2001, pp. 231-259.

[12] J. Timmis, "Artificial immune systems: A novel data analysis technique inspired by the immune network theory," Ph.D dissertation. Ceredigion, Wales: University of Wales, 2000.

[13] N. Kasabov, *Evolving Connectionist Systems: Methods and Applications in Bioinformatics, Brain Study, and Intelligence Machines*: Springer, 2002.

[14] N. Kasabov, "Evolving fuzzy neural networks for supervised/unsupervised on-line, knowledge-based learning," *IEEE Trans. on Man, Machine and Cybernetics, Part B: Cybernetics*, vol. 31, pp. 902-918, Dec. 2001.

[15] L. N. d. Castro and J. Timmis, *Artificial Immune Systems: A New Computational Approach*: Springer-Verlag, London. UK, 2002.

[16] L. N. D. Castro and F. J. V. Zuben, "An Evolutionary Immune Network for Data Clustering," in Proceedings of IEEE Brazilian Symposium on Artificial Neural Networks, Rio de Janeiro, 2000.

[17] The RIPE Routing Information Services. http://www.ris.ripe.net.

[18] H. S. Javitz and A. Valdes, "The NIDES statistical components: description and justification," SRI Network Information Center March 1993.

[19] BBC, Virus-like attack hits web traffic.http://news.bbc.co.uk/hi/technology/2693925.stm.

[20] People Daily, China Internet attacked by worm virus" January 2003. http://english.people.com.cn/200301/27/eng20030127_110810.shtml.