ANOMALY DETECTION IN MULTIDIMENSIONAL DATA USING NEGATIVE SELECTION ALGORITHM

Dipankar Dasgupta and Nivedita Sumi Majumdar
Division of Computer Science
Department of Mathematical Sciences
University of Memphis, TN

Abstract: While dealing with sensitive personnel data, the data have to be maintained to preserve its integrity and usefulness. The mechanisms of the natural immune system are very promising in this area, it being an efficient anomaly or change detection system. This paper reports some anomaly detection results with single and multidimensional data set using the Negative Selection Algorithm developed by Forrest [1].

1 INTRODUCTION

Data integrity is often compromised by undesirable changes, which need to be detected and corrected in a timely fashion. The human users may corrupt the data by accident or malice. Data may also be corrupted by a normal deterioration of the storage media. The occurrence of faults clearly cannot be avoided. Data can be validated, and such mistakes can be repaired using backup resources. However, this is an expensive solution and it may not be completely satisfactory. Many off-the-shelf database products in the market claim competence in dealing with such problems [2]. However, they generally perform basic integrity maintenance task – they can do type and range checking. With these products, any further specificity may become increasingly expensive. And there is no way the system can be taught the semantics of the data.

It is becoming increasingly clear that the conventional approaches take a limited view of the anomaly detection problem. The system's definition of the normal behavior usually changes with time. A set of values may be acceptable at a certain period of tine but at another time they may raise an alarm. For example, a sailor's frequency of a certain credit card usage during his offshore duty period is normal to be low. Where as when he is on shore, a higher frequency is acceptable. This nature of data pattern prevents the ordinary schemes of monitoring (like threshold of values etc.) from being very effective. Shifting of focus between different aspects of the data is also not possible without changing the design of the existing databases. Thus, the initial allocation of importance to the different attributes of the data is according to the user's initial requirements and at a later date a more detailed structure for any particular area is difficult to devise.

From the above discussion, one can see that an appealing solution is to set up a monitor, which can detect any discrepancy in the collected data based on its knowledge of the existing database. The system should be flexible in what fields it monitors and able to adapt to what it learns as the normal behavior shifts from time to time. It is desirable that the parameters of the tool are tunable, and is able to handle most data types uniformly.

2 IMMUNITY BASED APPROACH

In the recent years, a novel approach has begun to emerge which is the use of concepts from immunology to solve the above challenge [3, 4, 5]. The human immune system has a very distributed and adaptive, novel pattern recognition mechanism. An approximate modeling of some immunological metaphors has the potential to provide the answers we seek in anomaly detection. The body recognizes its own cells from those of the invaders, which is basis for the negative selection algorithm [1]. In other words, the system is be able to make self /non-self discrimination. The immune system uses learning, memory, and associative retrieval to solve recognition and classification tasks [7]. In particular, it learns to recognize relevant patterns, remember patterns that have been seen previously, and uses combinatorics to construct pattern detectors efficiently. Also, the overall behavior of the system is an emergent property of many local interactions. These remarkable information-processing abilities of the immune system provide several important inspirations to the field of computation [8].

The body creates antibodies randomly, out of which only those that do not match up with self cells are allowed to survive after a process of maturation. Thus, the antibodies in the bloodstream are those that can bind with foreign antigens for which it is designed. Though the antibody response is specific to an antigen, at the same time the antibody set accomplishes repertoire completeness. This is because they do not look for an exact match, which would be almost impossible to cover, as the non-self data search space is presumably very big. Cloning and hypermutation maintain the diversity of the antibody set. The antibodies are present through out the body without any central control and thus defend the body by this interaction in a distributed fashion. The immune system has an immunological memory in which previous encounters are learnt and remembered. Thus the system is capable of adjusting to newer definitions of the normal behavior.

Based on the above immunological principles, Forrest [1] developed a change detection technique called the Negative Selection Algorithm. In this work we extended the application of the algorithm to multidimensional data.

3 NEGATIVE SELECTION ALGORITHM

This approach is inspired by the information-processing properties of natural immune system. Animal immune systems are capable of distinguishing virtually any foreign cell or molecule from the body's own cells – this is known as the self/non-self discrimination problem. During generation of T-Cells, a pseudo-random genertic rearrangement process

0-7803-7282-4/02/$10.00 ©2002 IEEE

makes receptors. Then they undergo a censoring process, called negative selection, in the Thymus where T cells that react against self-proteins are destroyed; so only those that do not bind to self-proteins are allowed to leave the thymus. The basic principle is as follows:

- Define Self as a normal pattern of activity or stable behavior of a system/process, which needs to be monitored. In particular, we build up a database of normal behavior of each process of interest, and then logically split the pattern sequence to represent as a multiset S of equal size strings of length l over a finite alphabet.
- Generate a set R of detectors, each of which fails to match any string in S. We use a partial matching rule, in which two strings match if and only if they are identical in at least r contiguous positions, where r is a suitably chosen parameter.
- Monitor new observations (of S) for changes by continually matching the detectors against the representative of S. If any detector ever matches, a change (or deviation) must have occurred in the system behavior.

This algorithm generates the detectors for the r-contiguous bits matching rule. It has been adapted from Dasgupta et al [5]. Detector sets are generated using a linear time algorithm with respect to size of self. The algorithm has two phases. First it employs a dynamic programming technique to count recurrences in order to define an enumeration of all unmatched strings (i.e. feasible detectors). Second, a random subset of this enumeration is chosen to generate a detector set. In other words, given a collection of self strings S and the matching threshold r, the first phase of the algorithm determines the total number of unmatched string that exist for the defined self (S); then in the second phase, some of them are selected to generate a diverse set of detectors for monitoring data patterns (representative of the self).

The algorithm has been applied to anomaly detection in one dimensional time series data [6]. In this paper we follow the same algorithm and report some results with multidimensional personnel data. Personnel data has many attributes, which are categorical in nature. But we have only concentrated on the numeric attributes.

4. DEALING WITH MULTIDIMENSIONAL DATA

4.1 Multidimensional Data

We used a personnel dataset that is inherently multidimensional and typically have hundreds of attributes per record. We reduced the data dimensionality using Principal Component Analysis (PCA). It may be noted that, though this data is multidimensional, where the number of fields might run to hundreds, we can only apply PCA to a set of related fields because it may not be useful to apply PCA on uncorrelated data. This is not a disadvantage because at most times it will be necessary to look at only some of the fields to detect anomalous patterns. Some data, which are uncorrelated to other fields, may be looked at in one dimension. For example, it may be that all members classified to a particular group will have a similar numeric evaluation profile. In this case, we can look at that evaluation profile number in one dimension only to detect an anomaly. Also, these standards, for example the evaluation point threshold etc is likely to be time-sensitive. Thus it should up to the user to manipulate the values based on current needs and not hard-coded. The proposed technique can be very useful in this situation. It has all the right properties – a changing definition for normalcy with which the system must adapt and learn new things continuously.

4.2 Categorical Data

Personnel data records have many categorical fields. This posed a challenge to us and we continue to investigate the possibilities for this. It is difficult to numerically represent categorical data. Any attempt to do so arbitrarily imposes an ordering in the data, which is not true in real life. To apply the negative Selection Algorithm we need numerical data. In our experiments, we have chosen those personnel data attributes that are inherently numerical to bypass this problem.

4.3 Implementation Details

The algorithm for one-dimensional detection was extended for multi dimensional data. We assumed that our detection algorithm should work on any set of binary strings. The multidimensional data file was first passed through a Principal Component Analyzer. Discarding data fields that were responsible for less than 20% variability for the records, the dataset was reduced to two dimensions. This two-dimensional real-valued dataset was used to run the anomaly detection experiments. Each term of the pair, per record, was binary encoded using the same scheme as before. Next, these binary strings were Gray Coded.

Gray Coding was necessary here because in Binary Coding sometimes the Hamming Distance between two successive numbers like 7 and 8 is high. Which means that as we were going to use the r-contiguous matching rule, the actual closeness of two data values would not be appropriately reflected in a Binary encoding scheme. Whereas in Gray Encoding, two successive values differs in at most 1 bit position, and this would be useful. The two Gray Encoded strings thus obtained were simply concatenated into a single string and written onto a file. Result sets with both Binary and Gray Encoding schemes have been presented. The difference here was surprisingly not significant. The reason is probably because, by the very nature of our Binary encoding scheme, the actual nearness of two numbers is preserved. No windowing has been used here.

Windowing for time series data makes sense as we want to capture the gradual change in the scene, but here each real value is supposed to be absolutely independent of the other, which means that windowing is not sensible here. The results obtained are shown in fig. 9.

5. INTERFACE DESIGN

We developed a user friendly GUI to run the experiments. Figure 1 is a snap shot of the main screen. It exhibits the fields used in the experiments such as Enlistment type, Number of years of education, Major, Test score 1, Test score 2 and Trainee status. It gives the user choices to run experiments in multidimensional data projected to one or two dimensions with the two buttons labeled ' Experiments in One Dimension' and ' Experiments in Two Dimension' respectively.

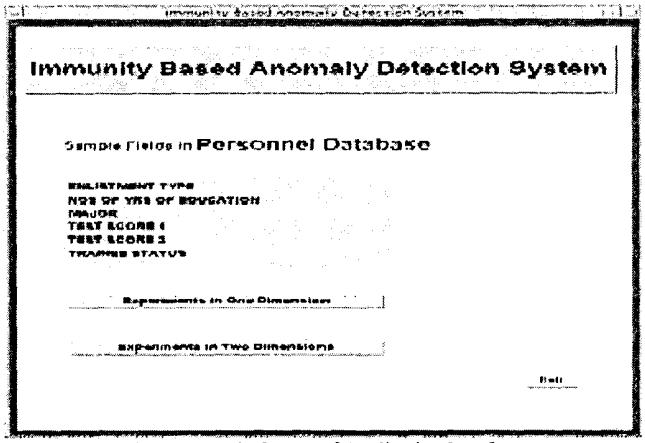


Figure. 1. Main Screen of Application Interface

Figure 2 shows the interface for one-dimensional experiments and a brief discussion of the parameters is presented here.

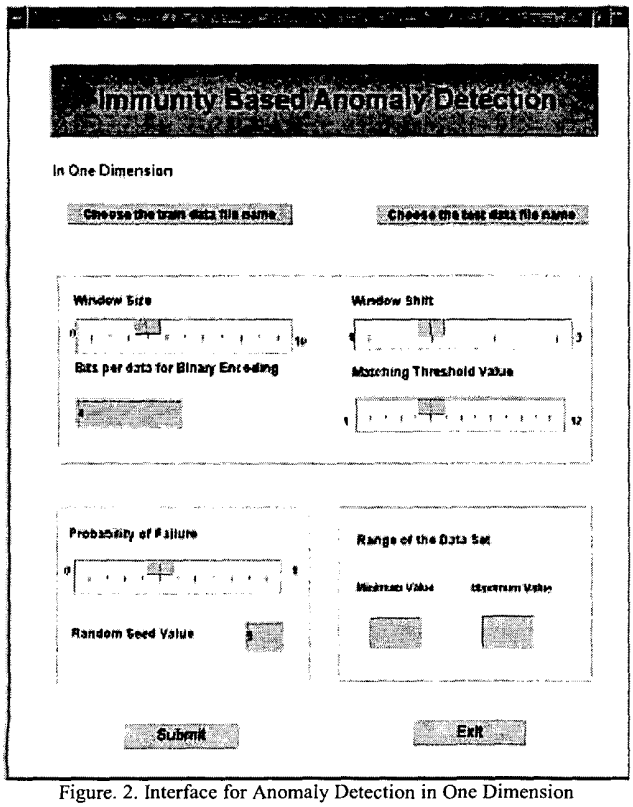


Figure. 2. Interface for Anomaly Detection in One Dimension

Bits per data: This will dictate the degree of numerical precision with which real numbers are represented in binary form. For example, 5-bit data encoding gives 30 intervals into which the range [Min, Max] of data is divided.

Minimum / Maximum Value: Min / Max Value of data.

Window Size: The number of samples encoded in a single pattern (each string in self).

Window Shift: The number of samples by which one pattern is shifted from the previous one in a moving window.

Matching Threshold: The r- value in the r- contiguous bit-matching rule is taken as input.

Probability of Failure: The size of detector set is influenced by this value. The detector set is a random selection from the exhaustive enumeration of the search space. And how many such feasible detectors are chosen is determined from this value.

Random seed value: This is the seed value for the random number generator. Figure 3 shows the screen shot for the interface to run the experiments in two-dimensional case.

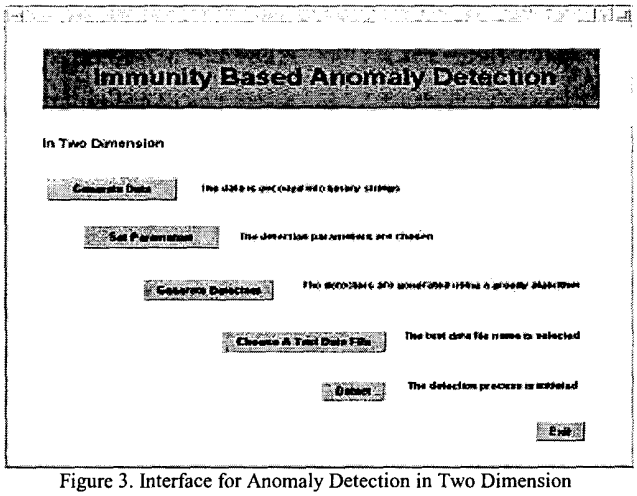


Figure 3. Interface for Anomaly Detection in Two Dimension

The button 'Generate Data' brings up the screen in Figure 4. We can choose to run the experiments in one, two or three dimensions. Note that all data handled here, are basically six dimensional and reduced by PCA to lower dimensions as discussed earlier.

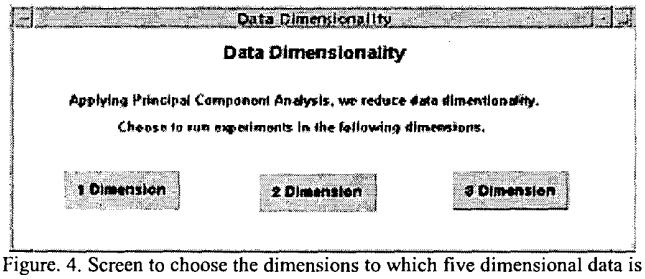


Figure. 4. Screen to choose the dimensions to which five dimensional data is to be reduced by PCA

The button 'Parameter Setting' brings up the screen in as shown in Figure 5. The matching threshold 'r' has to be

between 1 and the length of the binary string. In the above case, the total length has been set to 16. Since the five-dimensional original data set was passed through a Principal Component Analyzer, it was reduced to 2 dimensions with 23% loss. Eight bits were used to encode each data point of the pair and thus we got a total length of 16. The various parameters used for experiments are discussed below.

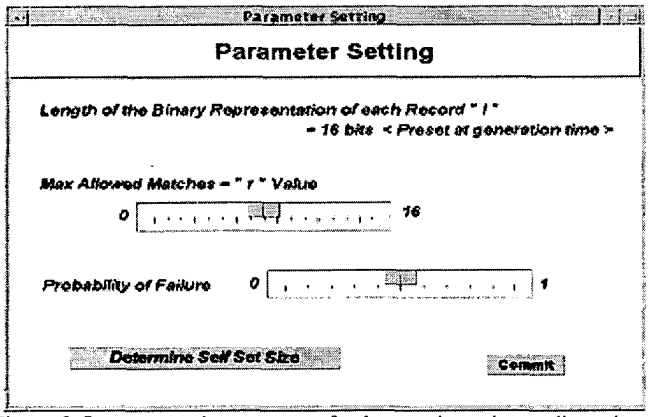


Figure. 5. Screen to set the parameters for the experiment in two dimensions

Probability of Failure: The probability of Failure value has to be set to constrain the number of detectors generated.

Determine Self-Size: The Determine Self Size button helps to auto detect the train data file size.

Commit: The Commit button stores the values thus set into a file, which is referred to by other parts of the programs to obtain the information.

Then, from the main screen again, we press the button 'Generate Detectors'. The detectors generated are stored in a file for future detection stage.

Next, from the main screen again, we press the button 'Choose a test data file' to select one. And the following screen appears for the user to make the selection.

Next, from the main screen again, we press the button ' Detect' to start the detection process. When it finishes it pops up the message stating how many antibodies were activated.

Identification/Personnel Data						
	1. Social Security Number					
	SSN	SSN verification status	SSN verification date	SSN change flag	...	Prev. SSN flag
	2. Name					
	Current Name	Previous Name	Name change date	...	Wage change status indicator	
	3. Personal data					
	...	...	...	...		...
	...	...	...	...		...
Education/Aptitude						
	Number of years of education		..	...	...	...
	1. Special Qualification Test Score					
	Sonar Test Memory Test Score		Nuclear Field Qualification Test Score		...	...
...						

TABLE 1. PERSONNEL DATA FORMAT

6 EXPERIMENTS & RESULTS

We have used some simulated data with the file structure inspired by the Navy Personnel Database. All personnel data is organized in files in a fixed format. There is an index file, which maintains the location of the data on the actual files. The data is collected and reported in a different order, but in this main file, they are arranged in their related order. Information are categorized into broad divisions as Identification/Personal Information, Service Data, Education/Aptitude, School History, Language Qualifications, Rate/Rating, Navy enlisted Classification override indicator, Dependents/Spouse Information etc. Then again all information regarding a sub-field, say the SSN of the Personal/Identification Information Section is grouped together. This ordering makes this file very valuable to search .for related information. A token representation is presented in Table 1.

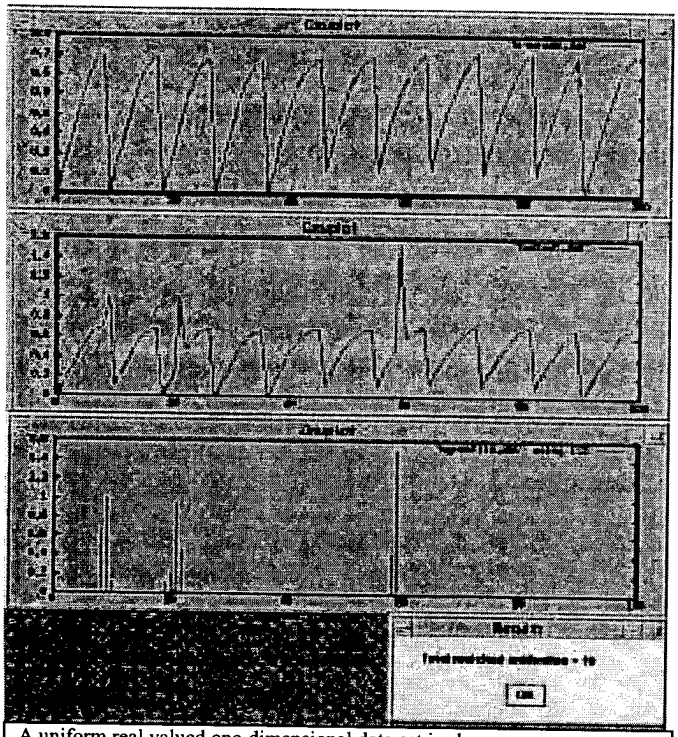
For experimentation, a five dimensional data set was generated. Imposing certain rules to make the data more realistic preserved the nature of the correlation that exists between the different fields. We then normalized each of the fields. Next we applied Principal Component Analysis to reduce data dimensionality.

We have used 3 bits to encode each of the five attributes except the 'major', for which we have used 6 bits. This means that for all other fields we can have 8 different values and for the 'major' field, we can have 64 different values. It is sufficient to consider only 8 distinct values for the above fields. A candidate's educational experience can be adequately specified by his claim to elementary schooling or high school diploma or two-year associate degree or four-year bachelor degree or a master's degree or a doctoral degree or post doctoral education or a specialized training. These 8 different stages are encoded in those eight admissible values. A candidate can obtain from $A\pm$ to $D\pm$ and here again we see, these eight distinct values can sufficiently express his test scores. Similar arguments can be furnished for the other attributes. So this adds up to a total string length of $3 + 3+ 6 + 3 + 3 + 3 = 21$ bits. Therefore we have a train file of data, which is converted, to a file of 21-bit records.

Our method of change detection works well with one-dimensional data. A result (fig. 6) is presented for illustration. This is time series data and this is the summary of the work by Dasgupta et al [4]. Many attributes like evaluation scores for a group is expected to display a particular pattern. This system can learn such a pattern and monitor unexpected alterations to it. Thus this part is important to our overall purpose and therefore the results are included here.

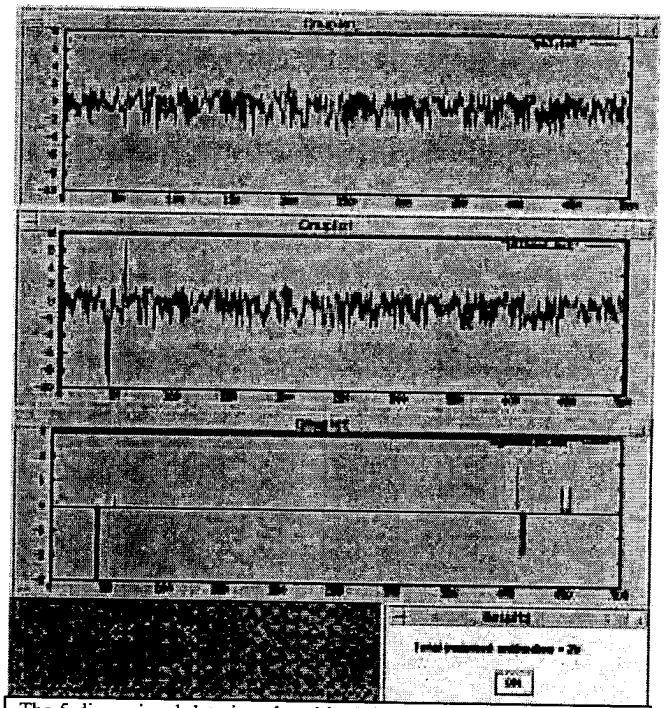
A result set obtained from five dimensional data reduced by PCA to two dimensions is given in figure 7. A result set obtained from five dimensional data reduced by PCA to one dimension is given in figure 8.

To add further proof to the proper functioning of the application, if the same file was input as test and train set, no antibodies were detected. This shows that the system would not raise alarm to what it has already been taught to expect as the 'normal'.



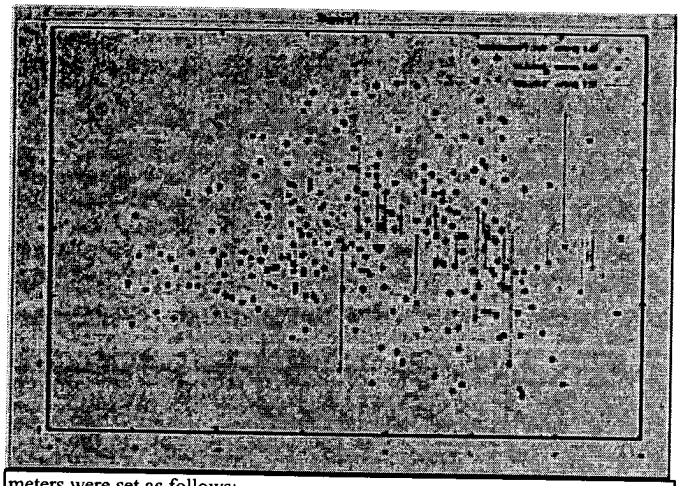
A uniform real valued one-dimensional data set is chosen.		
Matching threshold = 8	Window Size = 4	Random Seed = 5
Window Shift = 1	Prob. of failure = 0.2	Bits per data = 4
Size of training data set = 100	Size of testing data set = 100	

Figure. 6. Results for 1 dimensional data



The 5 dimensional data is reduced by PCA to 1 dimension. Results are shown above.		
Matching threshold = 10	Window Size = 4	Random Seed = 5
Window Shift = 1	Probability of failure = 0.2	Bits per data = 4
Size of training data set = 500	Size of testing data set = 500	

Figure. 8. Results for 5 dimensional data reduced by PCA to 1 dimension



meters were set as follows:		
Matching threshold = 8	Probability of failure = 0.2	
Size of training data set = 500	Size of testing data set = 500	Bits per data = 4

Figure. 7. Results for 5 dimensional data reduced by PCA to 2 dimensions

7. DISCUSSION OF RESULTS

Though the system manages to detect a reasonable number of anomalies, there are a few that go undetected. One possible reason may be the nature of the anomalous data. The "good" data was produced by imposition of certain rules. The "bad" data was produced by removal of those imposed rules. This means that some of the "bad" data may actually be conforming to the rules imposed and in that case, we do not actually want them to be detected at all. A better test would be to generate the "bad" data by reversing the rules. Further experimentations will be undertaken to investigate these issues.

Also, the data reduction by PCA usually destroys some information in the process. That is why all the anomalous data records (i.e. those that did not follow the integrity rules) were not detected by the system. Though when reduced to one dimension, the results appear to be promising, in fact all the anomalies introduced in the five dimensional test data sets were not apparent in the reduced dimension test data set. The observable ones got detected properly. It is clearly shown from One-Dimensional Experiment result set 5 to 6 (Table 2), where the detection rate has improved from 16.7% to 66.7%. The One-Dimensional Experiment result set 9 is also promising in that given the same test file, as was the training file there is no detection.

For the Two-Dimensional Experiment result sets; nowhere we have 100% success. This can be attributed to the fact that PCA does cause some information loss. That we have used two types of encoding schemes here, really do not make a significant difference as was previously expected. Again Two-Dimensional Experiment result set 9 is to show that there is no detection of self. These results clearly indicate that our novel method is workable and has potential for large-scale problem solving. Further research is necessary to investigate other data dimension reduction techniques like Independent Component Analysis. Also various immunological metaphors need to be explored.

Experiments in One dimension

Train File Name	Train parameters		Train Result	Test File Name (File Size)	Test Results:	
	Serial.	"r/l" val.	# Of detectors		# Of antibodies matched	Detection Rate
traindata.raw (Time series data) (512)*	1	9/16	413	testdata.raw (64)*	2	100%
	2	12/16	3946	testdata.raw (64)*	9	100%
Trainset.dat (Time series)	3	8/16	242	testset.dat (100)*	34	100%
	4	10/16	1009	testset.dat (100)*	37	100%
pt1.txt (5-D to 1-D) (500)*	5	8/16	116	pt1test.txt (500)*	17	16.7%
	6	10/16	802	pt1test.txt (500)*	45	66.67%
	7	10/16	802	pt1.txt (500)*	0	0.0%

Experiments in Two Dimension

Train File name	Train parameters				Train Results	Test File Name (Size)	Test Results:	
		"r/l" val.	Prob. Failure	Encoding	# Of detectors		# Of Abs matched	Detection Rate
pt2.txt	1	8/16	0.2	G	112	Pt2test.txt (500)*	23	36.7%
	2	10/16	0.2	G	285		31	76.7%
	3	12/16	0.2	G	1411		28	76.7%
	4	12/16	0.1	G	1769		35	86.7%
	5	8/16	0.2	B	107		29	40.0%
	6	10/16	0.2	B	288		31	73.4%
	7	1216	0.2	B	1413		40	90.0%
	8	1216	0.1	B	1768		41	93.0%
	9	12/16	0.2	B	1413	pt2.txt (500)*	0	0.0%

TABLE 2. EXPERIMENTAL RESULTS
G: GRAY ENCODING B: BINARY ENCODING

7. ACKNOWLEDGEMENTS

This work was supported by ONR (grant No. N00014-99-1-0721) and NSF (grant No. IIS-0104251) grants. We owe special thanks to Dr. Tanja Blackstone for her support and encouragement in pursuing this research and also providing us with the data format for personnel datasets. We would also like to thank our group members, Fabio Gonzalez and Jonathan Gomez in particular, for their help and useful comments.

8. REFERENCES

[1]. S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri. Self-nonself discrimination in a computer. In Proceedings of the IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, Los Alamitos, CA, pp. 202-212, 1994.

[2]. Salomon David. Data Compression: The Complete Reference. 2nd edition (October 2000) Springer Verlag.

[3]. Alexander Tarakanov and Dipankar Dasgupta, 'A formal model of an artificial immune system'. In the journal BioSystems, Vol. 55/1-3, pp. 151-158, February 2000.

[4]. Dipankar Dasgupta, 'Information Processing Mechanisms of the Immune System', A chapter in the book "New Ideas in Optimization" McGraw-Hill publication, 1999.

[5]. S. Forrest and S.A. Hofmeyr. "Immunology as information processing." In Design Principles for the Immune System and Other Distributed Autonomous Systems, edited by L.A. Segel and I. Cohen. Santa Fe Institute Studies in the Sciences of Complexity. New York: Oxford University Press (2000).

[6]. Dipankar Dasgupta and Stephanie Forrest, 'An Anomaly Detection Algorithm Inspired by the Immune System', Chapter 14 in the book entitled Artificial Immune Systems and Their Applications, Publisher: Springer-Verlag, Inc., pp 262-277, January 1999. pter in the book "New Ideas in Optimization" McGraw-Hill publication, 1999.

[7]. S. Forrest, B. Javornik, R. E. Smith and A. S. Perelson. Using genetic algorithms to explore pattern recognition in the immune system. In Evolutionary Computation 1:3, pp. 191-211, 1993.

[8]. Artificial Immune Systems and their applications – D. Dasgupta (Ed.) Springer – Verlag. 1999.