John Holland's Invisible Hand: An Artificial Immune System

Stephanie Forrest and **Steven A. Hofmeyr**
Dept. of Computer Science
University of New Mexico
Albuquerque, NM 87131-1386
{steveah,forrest}@cs.unm.edu

Abstract

We describe an artificial immune system (AIS) that is distributed, robust, dynamic, diverse and adaptive. It captures many features of the vertebrate immune system and places them in the context of the problem of protecting a network of computers from illegal intrusions. The AIS resembles a classifier system in many important ways. Similarities and differences are discussed.

1 INTRODUCTION

The parallel between immunology and classifier systems was noted as early as 1986 in [4]. In this work, a classifier system was used to model certain aspects of the immune system, by drawing an analogy between individual classifier rules and antibody types. Classifier strength represented the concentration of the antibody type, and interactions between classifier rules modeled Jerne's idiotypic network hypothesis [16]. Although the comparison was interesting, it had relatively little impact on classifier system research, and the two fields have continued to develop largely independently. Since that time, however, artificial immune system modeling has advanced significantly, and the emphasis has switched from idiotypic network theory to other aspects of immunology.

Likewise, in the twenty years since learning classifier systems were proposed [12], our thinking about cognitive systems has progressed in many interesting ways. One of the most significant changes has been the realization that living cognitive systems are situated in physical environments, and that their designs are both constrained and helped in important ways by this embedding. Brooks and others [8, 1] argue that it is fruitless to design intelligent systems in isolation from the environments in which they exist. Learning classifier systems were proposed as a general model of cognition, with a limited and highly abstract interface to the environment. We believe that research on classifier systems has suffered from this loose coupling with live environments. Situated intelligent artifacts are perhaps more complex to think about, because they cannot be neatly separated from their environments, but they can in some cases use their environments in ways that simplify their computations.

This paper describes an ongoing project to develop an artificial immune system (AIS), which is both closely related to classifier systems and embedded in a live environment. Our starting point for this line of research was a collection of pressing unsolved problems in computer security. Over the past several years we have designed and built prototypes for several computer security problems. Armed with that experience, we show here how to embed an architecture for adaptive behavior in a real-time environment with live agents (computers and the humans who operate them).

A more detailed description of the AIS architecture appears in [9, 10], together with experimental and analytical results of its performance. Our emphasis here is on the comparison with classifier systems.

2 THE IMMUNE SYSTEM

The immune system is highly complicated and appears to be precisely tuned to the problem of detecting and eliminating infections. It is also a compelling example of a distributed information-processing system, one which we can study for the purpose of designing better artificial adaptive systems.

The immune system is comprised of cells and molecules.[1] Recognition of foreign protein, called *antigen*, occurs when immune system detectors, including T cells, B cells, and antibodies, bind to antigen. Binding between detector and antigen is determined by the physical and chemical properties of binding regions on the cell surface. Binding is

[1] A good source for basic immunology is [14].

highly specific, so each detector recognizes only a limited set of structurally related antigen. When a detector and antigen bind, a complex set of events is initiated, often resulting in elimination of the antigen by scavenger cells called macrophages. (How antigen is bound and cleared depends on the type of detector involved.) A striking feature of the immune system is that the processes by which it generates detectors, identifies and eliminates foreign material, and remembers the patterns of previous infections are all highly parallel and distributed. This is one reason immune system mechanisms are so complicated, but it also makes them highly robust to failure of individual components and to attacks on the immune system itself.

Immunologists often describe the problem solved by the immune system as that of discriminating "self" from "other" (or "nonself") and eliminating other. In the language of classifier systems, the immune system must process external and internal messages, first classifying them as self or nonself, and in the case of dangerous foreign messages taking appropriate action. In immunology, self is generally taken to be the internal cells and molecules of the body, and nonself is any foreign material, particularly bacteria, parasites, and viruses. A more modern view emphasizes the immune system's role in eliminating infection in addition to its tolerance of self [17], (self vs. harmful other). As the ability to distinguish correctly between self and nonself is certainly crucial to the immune system's success, it is a common starting point for immune system models. Distinguishing between self and nonself is difficult for several reasons. First, the components of the body are constructed from the same basic building blocks, particularly proteins, as nonself. Proteins are an important constituent of all cells, and the immune system processes them in various ways, including in fragments called peptides which are short sequences of amino acids. Second, the size of the problem to be solved is large with respect to the available resources. For example, it has been estimated that the vertebrate immune system needs to be able to detect between 10^{11} and 10^{16} patterns [13], yet it has only about 10^5 different genes from which it must construct the entire immune system (as well as everything else in the body). The difficulty of this discrimination task is shown by the fact that the immune system can make mistakes. Autoimmune diseases provide many examples of the immune system confusing self with other.

3 THE ENVIRONMENT

There are compelling similarities between the problem faced by the human immune system and that of computer security. Both must protect highly complex, dynamically changing systems against intrusions from a wide variety of sources. Both must ensure the continued functioning of the system, and must ensure that the protective mechanisms do not seriously damage the system. However, these systems seem to have radically different ways of solving the problems confronting them. The immune system does so in a way that is distributed, flexible, adaptable, robust, degrades gracefully, and is resilient to errors and subversion [20]. These are properties we would like to see in computer security systems.

We have studied several computer security problems, including computer virus detection [6], host-based intrusion detection [5], and network security [9]. In this paper we concentrate on the latter—protecting a local-area broadcast network (LAN) from network-based attacks. Broadcast LANs have the convenient property that every location (computer) sees every packet passing through the LAN, so we can view the entire LAN as the "body" to be protected, and each computer on the LAN as a different location within it.

In this domain, we define self to be the set of normal pairwise connections (at the TCP/IP level) between computers, including connections between two computers in the LAN as well as connections between one computer in the LAN and one external computer (Figure 1). A connection is defined in terms of its "data-path triple"—the source IP address, the destination IP address, and the service (or port) by which the computers communicate. This definition of self, including the datapath triples, was introduced in [7, 18]. In our representation, this information is compressed to a single 49-bit string which unambiguously defines the connection. Self is then the set of normally occurring connections observed over time on the LAN, each connection being represented by a 49-bit string. Similarly, nonself is also a set of connections (using the same 49-bit representation), the difference being that nonself consists of those connections, potentially an enormous number, that are not normally observed on the LAN.

More generally, we can think of both the protected system (self) and infectious agents (nonself) as dynamically changing sets of bit strings. In cells of the body the profile of expressed proteins (self) changes over time, and likewise, we expect our set of protected strings to vary over time. Similarly, the body is subjected to different kinds of infections over time; we can view nonself as a dynamically changing set of strings.

4 ARCHITECTURE OF THE AIS[2]

Natural immune systems consist of many different kinds of cells and molecules—lymphocytes (B lymphocytes and T lymphocytes), macrophages, dendritic cells, natural killer cells, mast cells, interleukins, interferons, and many oth-

[2]The text in this section is excerpted from [10].

ers. Although these components have been identified and studied experimentally, it is not always well-understood what role they play in the overall immune response. In our AIS, we will simplify by introducing one basic type of detector cell which combines useful properties from several different immune cells. This detector cell will have several different possible states, roughly corresponding to thymocytes (immature T lymphocytes undergoing negative selection in the thymus), naive B lymphocytes (which have never matched foreign material), and memory B lymphocytes (which are long-lived and easily stimulated). The natural immune system also has many different types of effector cells, each implementing a different immune response (e.g., macrophage, mast cells, etc.), which we do not currently include in our model.

Each detector cell is represented by a single bit string of length $l = 49$ bits, and a small amount of state (see Figure 1). In effect, we are representing only the receptor region on the surface of a lymphocyte, or in the case of antibody molecules, the variable region of the molecule. It is this region that *binds* to foreign material, a process that we call recognition. There are many ways of implementing the detectors, for example, a detector could be a classifier, production rule, a neural network, or an agent. We chose to implement detection (binding) as string matching, where each detector is a string d, and detection of a string s occurs when there is a match between s and d, according to a *matching rule*. We use string matching because it is simple and efficient to implement, and easy to analyze and understand. Obvious matching rules include Hamming distance, edit distance, or the 1,0,# matching rule for classifiers. We chose a more immunologically plausible rule, called *r-contiguous bits* [19].

Two strings d and s match under the r-contiguous bits rule if d and s have the same symbols in at least r contiguous bit positions. The value r is a threshold and determines the specificity of the detector, which is an indication of the number of strings covered by a single detector. For example, if $r = l$, the matching is completely specific, that is, the detector will detect only a single string (itself; recall that l is the length of the detector bit string). A consequence of a partial matching rule with a threshold, such as r-contiguous bits, is that there is a trade-off between the number of detectors used, and their specificity: As the specificity of the detectors increases, so the number of detectors required to achieve a certain level of coverage also increases.

The detectors are grouped into sets on the LAN, one set per machine, or host; each host loosely corresponds to a different location in the body[3]. Because of the broadcast assumption, each detector set is constantly exposed to the current set of connections in the LAN, which it uses as a dynamic definition of self (i.e., the observed connections in a fixed time period are analogous to the set of proteins expressed in the thymus during some period of time). Within each detector set, new detectors, or thymocytes, are created randomly and asynchronously on a continual schedule, similar to the natural immune system. These new detectors remain *immature* for some period of time, during which they have the opportunity to match any current network connections. If a detector matches when it is immature, it is killed (deleted). This process is called *negative selection* [6], and closely resembles the negative selection of immature T lymphocytes (thymocytes) in the thymus. A potential problem with this scheme is that a nonself packet arriving during negative selection could cause immature detectors to be erroneously eliminated. However, if we assume that nonself packets are rare (a reasonable assumption), there are likely to be other mature detectors present to detect the foreign packet. We thus have a small loss of efficiency, from needlessly deleting a valid detector, but no appreciable loss of function.

Detectors that survive this initial testing phase are promoted to mature detectors (analogous to mature T lymphocytes leaving the thymus and mature B lymphocytes leaving the bone marrow). Each mature detector is now a valid detector that acts independently. If a mature detector d matches a sufficient number of packets (see activation threshold below), an alarm is raised. The time for which d is a naive B lymphocyte can be thought of as a learning phase. At the end of the learning phase, if d has failed to match a packet it is deleted, but if it has matched a sufficient number of nonself packets, it becomes a memory detector with a greatly extended lifetime. Memory detectors have a lower threshold of activation (see below), thus implementing a "secondary response" that is more sensitive and responds more aggressively than naive detectors to previously seen strings. Although these memory detectors are desirable, a large fraction of naive detectors must always be present, because the naive detectors are necessary for the detection of novel foreign packets, i.e. they are essential to anomaly detection.

Both the natural immune system and our AIS face the problem of "incomplete self sets." When T lymphocytes undergo negative selection in the thymus, they are exposed to most but not all of the proteins in the body. Consequently, the negative selection process can be incomplete in the sense that a lymphocyte could survive negative selection but still be reactive against a legitimate self protein (one that was not presented in the thymus) potentially leading to an auto-immune reaction. In our AIS, such an auto-

[3]The ability of immune system cells to circulate throughout the body is an important part of the immune system that we are currently ignoring. In our system, detectors remain in one location for their lifetime.

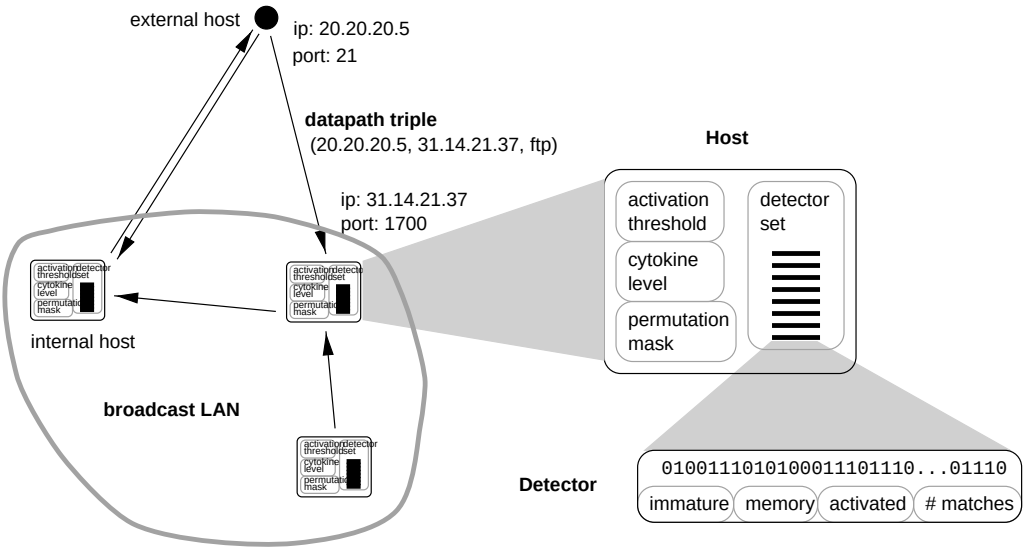

Figure 1: Architecture of Artificial Immune System.

immune reaction is called a *false positive*. False positives arise if we train the system on an incomplete description of self, and then encounter new but legitimate patterns. We would like the system to be tolerant of such minor, legitimate new patterns, but still detect abnormal activity, and we have implemented two methods designed to overcome this problem: Activation thresholds and sensitivity levels.

Activation thresholds are similar in function to avidity thresholds in lymphocytes. A lymphocyte is covered with many identical receptors, and it is only activated when sufficiently many receptors are bound to pathogens, i.e. when the avidity threshold for binding is exceeded. Analogously, each detector in the AIS must match multiple times before it is activated. Each detector records the number of times it matches, and it raises an alarm only when the number of matches exceeds the activation threshold, which is stored locally for each detector set. Once a detector has raised an alarm, it returns its match count to zero. This mechanism has a time horizon: Over time the count of matches slowly returns to zero. Thus, only repeated occurrences of structurally similar and temporally clumped strings will trigger the detection system.

However, some attacks may be launched from many different machines, in which case the first method is unlikely to be successful. To detect such distributed coordinated attacks, we introduce a second method, called *sensitivity level* (labeled *cytokine level* in Figure 1). Whenever the match count of a detector goes from 0 to 1, the local activation threshold is reduced by one. Hence, each different detector that matches for the first time "sensitizes" the detection system, so that all detectors on that machine are more easily activated in the future. This mechanism also has a time horizon; over time, the activation threshold gradually returns to its default value. Thus, this method will detect diverse activity from many different sources, provided that activity happens within a certain period of time. This mechanism roughly captures the role that inflammation, cytokines, and other molecules play in increasing or decreasing the sensitivity of individual immune system lymphocytes within a physically local region.

Negative selection and the maturation of naive cells into memory cells are two simple learning mechanisms used by the immune system. A third form of immune-system learning, one that resembles a genetic algorithm (without crossover), is incorporated into our model—affinity maturation. In its simple form, detectors compete against one another for foreign packets, just as lymphocytes compete to bind foreign antigen. In the case where two detectors simultaneously match the same packet, the one with the closest match (greatest fitness) wins, similarly to bidding in classifier systems. This introduces pressure for more specific matching into the system, causing the system to discriminate more precisely between self and nonself. We propose, although we have not yet implemented this, that successful detectors (those that bind many foreign packets) will undergo proliferation (making copies and migrating to other computers) and somatic hypermutation (copying with a high mutation rate).

The concept of a *second signal*, known as *co-stimulation*, is often used to explain certain immunological responses. One example of a second signal is a T-helper lymphocyte. When a B lymphocyte (that is possibly a mutated

descendant of an earlier lymphocyte that survived negative selection) binds a foreign peptide (the first signal), it requires a T-helper lymphocyte (that has been censored against self in the thymus) in order to trigger an immune response. This second-signal system prevents mutating B-lymphocyte lines from incorrectly reacting against self. In our system, we use a human as the second signal. When a detector raises an alarm, there is some chance that it is a false alarm (auto-immune reaction). Before taking action, the AIS waits a fixed amount of time (say 24 hours) for a co-stimulatory signal, which in the current implementation is an email message from a human. If the signal is received (confirming the anomaly), the detector enters the competition to become a memory detector, but if it loses the competition, it remains naive and has its match count reset to 0. If the second signal is not received, the AIS assumes that it was a false alarm and destroys the detector (as in the natural immune system).

It might seem more natural to send messages to the AIS in the case of false alarms instead of true anomalies, so that the AIS can adjust itself appropriately by immediately deleting the auto-reactive detectors. Unfortunately, this would create a vulnerability, because a malicious adversary could send signals to the AIS, labeling true foreign packets as false alarms, thus tolerizing the AIS against certain forms of attack. The form of co-stimulation that we have used is much more difficult to subvert. Because false alarms are expected to be more frequent than true anomalies, our co-stimulation method has the additional advantage that action by the human operator is required in the less frequent case.

Figure 1 summarizes the lifecycle of a detector. A detector is initially randomly created, and then remains immature for a certain period of time, which is the tolerization period. If the detector matches any string a single time during tolerization, it is replaced by a new randomly generated detector string. If a detector survives immaturity, it will exist for a finite lifetime. At the end of that lifetime it is replaced by a new random detector string, unless it has exceeded its match threshold and becomes a memory detector. If the activation threshold is exceeded for a mature detector, it is activated. If an activated detector does not receive costimulation, it dies (the implicit assumption is that its activation was a false positive). However, if the activated detector receives costimulation, it enters the competition (see above) to become a memory detector with an indefinite lifespan. Memory detectors need only match once to become activated.

Each of the mechanisms described above can be implemented with a single detector set running on a single location. We can trivially gain efficiency advantages by distributing the single detector set across all locations on the LAN, thus distributing the computational cost of intrusion detection. Such distribution will give linear speedup, because there are no communication costs (apart from the signaling of alarms and costimulation). However, we take advantage of another immune system feature to implement a more powerful form of distribution.

The protein *major histocompatibility complex* (MHC) plays an important role in immune systems, because it transports protein fragments (called peptides) from the interior regions of a cell to its surface, *presenting* these peptides on the cell's surface. This mechanism enables roving immune system cells to detect infections in cells without penetrating the cell membrane. There are many variations of MHC, each of which binds a slightly different class of peptides. Each individual in a population is genetically capable of making a small set of these MHC types (about ten), but the set of MHC types varies in different individuals. Consequently, individuals in a population are capable of recognizing different profiles of peptides, providing an important form of population-level *diversity*[4]. Our AIS uses permutation masks to achieve a similar kind of diversity. A permutation mask defines a permutation of the bits in the string representation of the network packets. Each detector set has a different, randomly-generated, permutation mask. One limitation of the negative-selection algorithm as originally implemented is that it can result in undetectable abnormal patterns called holes, which limit detection rates [3, 2]. Holes can exist for any symmetric, fixed-probability matching rule, but by using permutation masks, we effectively change the match rule on each host, and so overcome the hole limitation. Thus, the permutation mask controls how the network packet is presented to the detection system, which is analogous to the way different MHC types present different sets of peptides on the cell surface.

The discussion thus far has concentrated on the detection side of our AIS and ignored questions of immune response. When stimulated by lymphocytes bound to the cell surface, immune system cells secrete a variety of molecules known collectively as *cytokines*. These cytokines diffuse from the site where they were secreted, and in turn play a role in stimulating or suppressing other immune system cells. Thus, cells that detect pathogens can communicate using these molecular signals with cells that assist in eliminating the pathogens (e.g., mast cells, macrophages, etc.). Although we plan to extend our model in the future to include this kind of signaling and response, the current model eliminates this complication (except for the sensitivity level).

[4]For example, there are some viruses, such as the Epstein-Barr virus, that have evolved dominant peptides which cannot be bound by particular MHC types, leaving individuals who have those MHC types vulnerable to the disease [15].

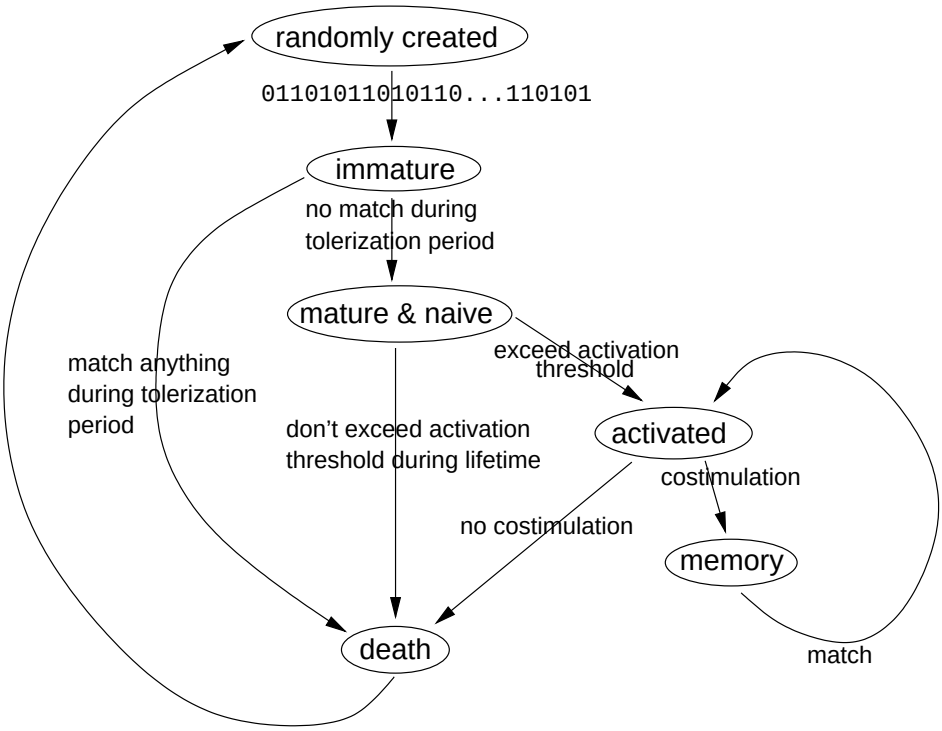

Figure 2: The Lifecycle of a Detector.

5 COMPARISON WITH CLASSIFIER SYSTEMS

The AIS outlined in Section 4 resembles the architecture of a classifier system[11], although most of the details are different (see Table 1). The mapping between classifier systems and our AIS is not 1-1, however. In this section, we point out both the similarities and differences.

Each detector d corresponds to the condition part of a classifier, where the match rule is r-contiguous bits instead of the traditional $1, 0, \#$ alphabet used in classifier systems. The parameter r is a measure of the specificity of the detectors, much like the number of don't cares in a classifier condition is a measure of its generality. In the current AIS, there is nothing corresponding to the action part of a classifier rule. However, if we concatenate some bits to each detector to specify a response (analogous to different antibody *isotypes*), then each immune cell (detector plus response bits) would correspond quite directly to the condition/action rule format of classifier systems. Less directly analogous are activation thresholds, which roughly correspond to Holland's proposal for *support*, and sensitivity levels which serve a similar role to message *intensity*. In both cases, the AIS mechanism is quite different from that used in classifier systems, but the reason for the mechanism is similar—in the one case to aggregate information from multiple sources and in the second case to vary the sensitivity of the system dynamically. Both activation thresholds and sensitivity levels decay over time, similarly to the role of tax in classifier systems.

In place of the message list we have a continuous flux of datapath triples that represent the current state of the environment. Currently, the only network connections generated by the AIS (analogous to internally generated messages in a classifier system) are those resulting from alarms being sent to the human operator.

There is no direct analog of the negative-selection algorithm in classifier systems, except the learning rules (such as genetic algorithm and trigger conditions) under which new rules are generated. Bidding for messages in classifier systems is analogous to immune cells competing to bind to foreign datapaths. Likewise, we introduce pressure for specificity, which is reminiscent of classifier systems, by allowing the more specific match to win the competition.

The role of the bucket brigade (credit assignment) and the genetic algorithm is played by our affinity maturation model of learning, although ours is simpler in the sense that we assign credit directly from the environment to the detectors, and do not pass strength among immune cells. A more direct analog of the bucket brigade would occur if we tried to build up idiotypic networks of immune cell

Classifier Systems	Artificial Immune System
classifier condition	detector
classifier action	isotypes
$1, 0, \#$ matching	r-contiguous bits
classifier strength	immature, mature, activated, and memory states
message list	network traffic (datapath triples)
competition for packets	bidding for messages
more specific match wins	more specific match wins
support	activation threshold
message intensity	sensitivity level
bucket brigade	affinity maturation
genetic algorithm, triggering	random detectors, negative selection
?	permutation masks

Table 1: Tentative comparison of artificial immune system with classifier systems

in which immune cells stimulate and repress other immune cells, as Jerne proposed [16]. Although this is appealing from an adaptive design perspective, there is little if any experimental evidence that such networks exist in natural immune systems. In classifier systems, each classifier's *strength* is represented by a real number. A classifier's strength determines the probability of it being deleted or replicated through the genetic algorithm. In the AIS, each detector is in one of several discrete states: Immature, mature, activated, or memory. Which state it is in determines the likelihood of it being deleted, replicated, or mutated. Note, in the current system, only the first option is implemented.

The AIS is essentially a stimulus/response system, where the stimuli are network packets, classification of inputs does not involve a large amount of internal processing, and the response is an email message to a human operator. The natural immune system is considerably more complicated, with highly complex internal regulatory mechanisms and several different kinds of potential responses. The regulatory mechanisms appear to be implemented through signaling molecules such as cytokines (discussed earlier). Our plan is to incorporate internal feedbacks and self-regulation by extending the cytokine system (we saw a primitive form of this in the sensitivity level).

Permutation masks have no direct analog in classical classifier systems. However, they do provide a natural partitioning of the set of detectors, something that has eluded classifier systems. We speculate that different detector sets might discover different kinds of regularities in network traffic (due to the combination of permutation with the locality of the r-contiguous bits matching rule.

6 CONCLUSION

In the previous sections we described an architecture for an adaptive artificial system based on the immune system. It incorporates several important immune-like properties, including detection of novel foreign patterns (because it is an anomaly detector), distributed detection via the negative-selection algorithm, and diversity across individuals (computers) in a population (the protected network) using permutation masks. It incorporates several forms of adaptation on different time scales, and it addresses an important problem of practical significance (network intrusion detection). Although the mapping between classifier systems and the AIS is not $1-1$, we believe that the system we have described captures many of the important properties of classifier systems and provides an interesting point of comparison.

Most of the features described in this paper have been implemented in a software prototype, which we have tested in the CS Dept. at UNM. It has discovered outside attacks as well as interesting anomalies generated internally. In one recent experiment, consisting of 50 computers on a switched subnet, with 100 detectors on each computer (each detector consisting of a 49-bit string), we detected 100% of the eight abnormal incidents we tested against and achieved a false-alarm rate of about two per day. This compares with millions of false alarms per month that have been reported anecdotally for some fielded systems.

Moving beyond the computer network intrusion-detection application that we have described, the AIS might be applied to other classes of networks, including social networks, organizations, networks of markets, neurological networks, or ecological networks. Like our LAN with external connections, these networks consist of many components that are sparsely connected, in which there are some ordered and some random components, and in which the exact set of connections is not static. There are important computations associated with each of these networks, and they would provide an important test of the generality of our architecture in its ability to discriminate normal and abnormal activity and to respond appropriately.

Acknowledgments

The authors acknowledge the support of the Defense Advanced Research Projects Agency (grant N00014-96-1-0680), the National Science Foundation (grant IRI-9711199), the Office of Naval Research (grant N00014-99-1-0417), the IBM Partnership award, and the Intel Corporation.

References

[1] H. J. Chiel and R. D. Beer. The brain has a body: Adaptive behavior emerges from interactions of nervous system, body and environment. *Trends in Neurosciences*, 20:553–557, 1997.

[2] P. D'haeseleer. An immunological approach to change detection: Theoretical results. In *Proceedings of the 9th IEEE Computer Security Foundations Workshop*, Los Alamitos, CA, 1996. IEEE Computer Society Press.

[3] P. D'haeseleer, S. Forrest, and P. Helman. An immunological approach to change detection: Algorithms, analysis and implications. In *Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy*, Los Alamitos, CA, 1996. IEEE Computer Society Press.

[4] J. D. Farmer, N. H. Packard, and A. S. Perelson. The immune system, adaptation, and machine learning. *Physica*, 22D:187–204, 1986.

[5] S. Forrest, S. A. Hofmeyr, and A. Somayaji. A sense of self for unix processes. In *Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy*, Los Alamitos, CA, 1996. IEEE Computer Society Press.

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

[7] L. T. Heberlein, G. V. Dias, K. N. Levitt, B. Mukherjee, J. Wood, and D. Wolber. A network security monitor. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Press, 1990.

[8] H. Hendriks-Jansen. *Catching Ourselves in the Act*. MIT Press, Cambridge, MA, 1996.

[9] S. A. Hofmeyr. *An immunological model of distributed detection and its application to computer security*. PhD thesis, Univ. of New Mexico, Albuquerque, NM, 1999.

[10] S. A. Hofmeyr and S. Forrest. Immunity by design: An artificial immune system. In *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO)*, in press.

[11] J. H. Holland, K. J. Holyoak, R. E. Nisbett, and P. Thagard. *Induction: Processes of Inference, Learning, and Discovery*. MIT Press, 1986.

[12] J.H. Holland and J. Reitman. Cognitive systems based on adaptive algorithms. In D. Waterman and F. Hayes-Roth, editors, *Pattern-directed Inference Systems*. Academic Press, New York, 1978.

[13] J. K. Inman. The antibody combining region: Speculations on the hypothesis of general multispecificity. In G. I. Bell, A. S. Perelson, and Jr. G. H. Pimbley, editors, *Theoretical Immunology*, pages 243–278. M. Dekker, NY, 1978.

[14] C. A. Janeway and P. Travers. *Immunobiology: the immune system in health and disease*. Current Biology Ltd., London, 2nd edition, 1996.

[15] C. A. Janeway and P. Travers. *Immunobiology: The Immune System in Health and Disease, 3rd Edition*. Current Biology Ltd., London, 1996.

[16] N. K. Jerne. Toward a network theory of the immune system. *Ann. Immunol. Inst. Pasteur*, 125C:373–389, 1974.

[17] P. Matzinger. Tolerance, danger, and the extended family. *Annual Reviews in Immunology*, 12:991–1045, 1994.

[18] B. Mukherjee, L. T. Heberlein, and K. N. Levitt. Network intrusion detection. *IEEE Network*, pages 26–41, May/June 1994.

[19] J. K. Percus, O. E. Percus, and A. S. Perelson. Predicting the size of the antibody-combining region from consideration of efficient self/nonself discrimination. In *Procedings of the National Academy of Science 90*, pages 1691–1695, 1993.

[20] A. Somayaji, S. A. Hofmeyr, and S. Forrest. Principles of a computer immune system. In *Proceedings of the Second New Security Paradigms Workshop*, 1997.