Information processing in Artificial Immune Systems: approaches and state of the art

I.Antoniou, P.Avtonomov, D.Kapustin, D.Kozhevnikova, V.Lazarev, Yu.Melnikov

International Solvay Institutes for Phusics and Chemistry, Campus Plaine ULB, CP231, Bd.du Triomphe, Brussels, Belgium

The notion of Artificial Immune System(AIS) has appeared as an attempt to get hints from the nature for more effective information processing. Artificial immune systems (AIS) represent the cutting-edge of classification and data analysis tools belonging to the emerging field within artificial intelligence and computer science called *immunological computation.* They are similar to case-based reasoning (CBR), neural networks (NN), Kohonen networks, adaptive resonance networks (ART), and self-organising clustering systems in that they form a reduced or compiled "sense" of environmental awareness through observation of software or hardware stimulus.

The natural immune system is a complex network of organs containing cells that recognize foreign substances in the body and destroy them. It protects vertebrates against pathogens, or infectious agents, such as viruses, bacteria, fungi, and other parasites. The human immune system is the most complex.

Some of the properties of the immune system that might be of interest to a computer scientist are:

Uniqueness: the immune system of each individual is unique and therefore vulnerabilities differ from one system to the next.

Distributed detection: the detectors used by the immune system are small and efficient, are highly distributed, and are not subject to centralized control or coordination.

Imperfect detection: by not requiring absolute detection of every pathogen, the immune system is more flexible: the body can trade off resources used on protection for comprehensiveness of coverage.

Anomaly detection: the immune system can detect and react to pathogens that the body has never before encountered.

Learning and memory (adaptability): the immune system can learn the structures of pathogens, and remember those structures, so that future responses to the pathogens can be much faster.

These properties result in a system that is scalable, resilient to subversion, robust, very flexible, and that degrades gracefully.

This review represents the group of main researchers and research groups in the field of AIS.

****

Dr. Dipankar Dasgupta is an assistant Professor of Computer Science in Mathematical Sciences department at the University of Memphis, Tennessee. Dipankar Dasgupta's research interests are broadly in the area of scientific computing, tracking real-world problems through interdisciplinary cooperation. His areas of special interests include Artificial Intelligence, Genetic Algorithms, Neural Networks, Artificial Immune Systems. One of his current interests is the application of these intelligent techniques in computer and network security. According to publicly available information he is working on the following projects.

Project Title: Immunity-Based Intrusion Detection Systems (Funded by DARPA, Period: April 2000 - March 2003).

The goal of this project is to develop an intelligent multi-agent system for intrusion/anomaly detection and response in networked computers. The approach is inspired by the defense mechanisms of the immune system that is a highly distributed in nature. In this approach, immunity-based agents roam around the machines (nodes or routers), and monitor the situation in the network (i.e. look for changes such as malfunctions, faults, abnormalities, misuse, deviations, intrusions, etc.). These agents can mutually recognize each other's activities and can take appropriate actions according to the underlying security policies. Specifically, their activities are coordinated in a hierarchical fashion while sensing, communicating and generating responses. Moreover, such an agent can learn and adapt to its environment dynamically and can detect both known and unknown intrusions.

***Project Title: Anomaly Detection using a Technique Inspired by the Immune System
(Funded by ONR, Period: May 1999 - August 2001).***

We are investigating an immunity-based anomaly detection algorithm for monitoring significant changes in time series data and for data quality analysis. This anomaly detection algorithm incorporates probabilistic methods motivated by the negative selection mechanism of the immune system to detect deviations from historical behavior patterns. In particular, the detection system learns the knowledge of the domain from historical data set to generate probabilistically a set of pattern detectors that can detect any abnormalities in the behavior pattern of the monitored data series. Moreover, the detection system is distributed and dynamic (can be updated by generating a new set of detectors as the behavior shifts due to changes in organizational or operational environments). The adaptive nature of this algorithm makes it superior to the static approaches that in current practice.

Project Title: Preliminary Research on Immunity-Based Computational Techniques (NSF SGER grant, Period: March 2001 - February 2002).

The PI will conduct a preliminary investigation of immunity-based computational techniques to pave the way for more complex studies of this subject in the future. The ultimate goal of this research is to develop computational techniques inspired by the natural immune system for solving real-world science and engineering problems. The natural immune system is a distributed novel-pattern recognizer, which uses intelligent mechanisms to detect a wide variety of antigens (novel patterns). From the computational point of view the immune system uses learning, memory, and associative retrieval to solve recognition and classification tasks. The immune system is a subject of great research interest, not only in the hope of finding cures for many diseases but also as a means for understanding its powerful information processing capabilities. In the current project the PI will investigate immunological principles, explore the underlying concepts and mechanisms, and take initial steps towards the development of intelligent computational techniques for solving problems in the field of science and engineering.

****

A big group of researchers lead by Stephanie Forrest is working in University of New Mexico. It includes Hajime Inoue, Geoff Hunsicker, Anil Somayaji and Steven Hoffmeyr.

Here are the projects of this group.

Intrusion Detection Project

The target of this project is an intrusion-detection system for networked computers. Discrimination between normal and abnorml behavior must be based on some characteristic structure that is both compact and universal in the protected system. Further, it must be sensitive to most undesirable behavior. Most earlier work on intrusion detection monitors the behavior of individual users, but researchers decided to concentrate on system-level processes. They define self in terms of short sequences of system calls executed by privileged processes in a networked operating system. Preliminary experiments on a limited testbed of intrusions and other anomalous behavior show that short sequences of system calls (currently sequences of length 6) provide a compact signature for self that distinguishes normal from abnormal behavior.

The strategy for their intrusion-detection system is to collect a database of normal behavior for each program of interest. Each database is specific to a particular architecture, software version and configuration, local administrative policies, and usage patterns. Once a stable database is constructed for a given program in a particular environment, the database can then be used to monitor the program's behavior. The sequences of system calls form the set of normal patterns for the database, and sequences not found in the database indicate anomalies. In the first stage, they collect samples of normal behavior and build up a database of characteristic normal patterns (observed sequences of system calls). Parameters to system calls are ignored, and they trace forked subprocesses individually. In the second stage, they scan traces that might contain abnormal behavior, matching the trace against the patterns stored in the database. If a pattern is seen that does not occur in the normal database, it is recorded as a mismatch. In their current implementation, tracing and analysis are performed off-line. Mismatches are the only observable that we use to distinguish normal from abnormal. They observe the number of mismatches encountered during a test trace and aggregate the information in several ways.

Although this method does not provide a cryptographically strong or completely reliable discriminator between normal and abnormal behavior, it is much simpler than other proposed methods and could potentially provide a lightweight, real-time tool for continuously checking executing code. Another appealing feature is that code that runs frequently will be checked frequently, and code that is seldom executed will be infrequently checked. Thus, system resources are devoted to protecting the most relevant code segments. Finally, given the large variability in how individual systems are currently configured, patched, and used, we expect that databases at different sites would likely differ enough to give each protected location its own unique signature (. A unique signature is important for another reason---it could provide a form of identity that is much harder to falsify than, for example, an IP address

Network based intrusion detection

They are designing and testing a prototype distributed intrusion detection system (IDS) that monitors TCP/IP network traffic. Each network packet is characterised by the triple (source host, destination host, network service). The IDS monitors the network for the occurrence of uncommon triples, which represent unusual traffic patterns within the network. This approach was introduced by researchers at the University of California, Davis, who developed the Network Security Monitor (NSM), which monitors traffic patterns on a broadcast LAN. NSM was effective because most machines communicated with few (3 to 5) other machines, so any intrusion was highly likely to create an unusual triple and thus trigger an alarm.

Although successful, NSM has serious limitations. It is computationally expensive, requiring its own dedicated machine, and even then only being able to monitor existing connections every five minutes. Further, the architecture of NSM does not scale: The computational complexity increases as the square of the number of machines communicating. Finally, NSM is a single point of failure in the system because it runs on a single machine. These limitations can be overcome by distributing the IDS over all machines in the network. Distribution will make the IDS robust by eliminating the single point of failure and will make it more flexible and efficient; computation can vary from machine to machine, fully utilizing idle cycles.

The architecture of NSM is not easily distributable. Distributing NSM would require either excessive resource consumption on every machine upon which it was run, or communication between machines. The immune system has interesting solutions to a similar problem of distributed detection. The researchers have designed a distributed IDS based on the architecture of the immune system. This allows the IDS to function efficiently on all machines on the LAN, without any form of centralized control, data fusion or communication between machines. The architecture is scalable, flexible and tunable.

Their IDS depends on several "immunological" features, the most salient being negative detection with censoring, and partial matching with permutation masks. With negative detection, the system retains a set of negative detectors, that match occurences of abnormal or unusual patterns (in this case, the patterns are binary string representations of network packet triples). The detectors are generated randomly and censored (deleted) if they match normal patterns. Partial matching is implemented through a matching rule, which allows a negative detector to match a subset of abnormal patterns. Partial matching reduces the number of detectors needed, but can result in undetectable abnormal patterns called holes, which limit detection rates. We eliminate holes by using permutation masks to remap the triple representation seen by different detectors.

They have conducted controlled experiments on a simulation that uses real network traffic as normal, and synthetically generated intrusive traffic as abnormal. Using a system of 25 detectors per machine on a network of 200 machines, one out of every 250 nonself patterns goes undetected, which is a false-negative error rate of 0.004. This number is conversative because intrusions will almost always generate more than one anomalous pattern. The computational impact of 25 detectors per machine is negligible, so performance can be improved by using more detectors per machine: If the number of detectors is doubled to 50 per machine, the error rate reduces by an order of magnitude. These results indicate that holes can be effectively eliminated using permutation masks, and that consequently the IDS can provide comprehensive coverage in a robust, fully distributed manner.

Previously, in the 1996 IEEE Symposium on Security and Privacy, they reported a technique for intrusion detection using sequences of system calls. Although the vision here is the same, this current research differs in the domain of application (network traffic), and draws far more strongly on the immune analogy.

Distributed Change Detection

T cells are an important class of detector cells in the immune system. There are several different kinds of T cells, each of which plays its own role in the immune response. All T cells, however, have binding regions that can detect antigen fragments (peptides). These binding regions are created through a pseudo-random genetic process, which we can think of analogously to generating random strings. Given that the binding regions, called receptors, are created randomly, there is a high probability that some T cells will detect self peptides. The immune system solves this problem by sending nonfunctional T cells to an organ called the thymus to mature. There are several stages of T-cell maturation, one of which is a censoring process in which T cells that bind with self proteins circulating through the thymus are destroyed. T cells that fail to bind to self are allowed to mature, leave the thymus, and become part of the active immune system. This process is called negative selection .Once in circulation, if a T cell binds to antigen in sufficient concentration, a recognition event can be said to have occurred, triggering the complex set of events that leads to elimination of the antigen.

The T cell censoring process can be thought of as defining a protected collection of data (the self proteins) in terms of its complementary patterns (the nonself proteins). We have used this principle to design a distributable change-detection algorithm with interesting properties. The algorithm works by first defining a set of data or activity patterns to protect (called self), then generating detectors that fail to match self, and finally, using the detectors to monitor self for changes. Details of these steps are given in our papers.

The algorithm has several interesting properties. First, it can be easily distributed because each detector covers a small part of nonself. A set of negative detectors can be split up over multiple sites, which will reduce the coverage at any given site but provide good system-wide coverage. To achieve similar system-wide coverage with positive detection is much more expensive: either a nearly complete set of positive detectors will be needed at every site, resulting in multiple copies of the detection system, or the sites must communicate frequently to coordinate their results. A second point about this algorithm is that it can tolerate noise, depending on the details of how the matching function is defined. Consequently, the algorithm is likely to be more applicable to dynamic or noisy data like the intrusion-detection example than, for instance, in cryptographic applications where efficient change-detection methods already exist.

Diversity to Reduce Vulnerability

In biological systems, diversity is an important source of robustness. A stable ecosystem, for example, contains many different species which occur in highly-conserved frequency distributions. If this diversity is lost and a few species become dominant, the ecosystem becomes susceptible to perturbation s such as catastrophic fires, infestations, and disease. Similarly, health problems often emerge when there is low genetic diversity within a species, as in the case of endangered species or animal breeding programs. The vertebrate immune system offers a third example, providing each individual with a unique set of immunological defenses, helping to control the spread of disease within a population. Computers, by contrast, are notable for their lack of diversity. Manufacturers produce multitudes of identical copies from a single design, with the goal of making every chip of a given type and every copy of a given program identical.

As computers increasingly become mass-market commodities, the decline in the diversity of available hardware and software is likely to continue, and as in biological systems, such a development carries serious risks. All the advantages of uniformity become potential weaknesses when they can be exploited by an attacker, because once a method is created for penetrating the security of one computer, all computers with the same configuration become similarly vulnerable. The potential danger grows with the population of interconnected and homogeneous computers.

If every intrusion, virus, or worm had to be explicitly crafted to a particular machine, the cost of trying to penetrate computer systems would go up dramatically. We are studying methods for introducing diversity that focus on unnecessary consistencies. Each aspect of a programming language that is "arbitrary" or "implementation dependent" is an opportunity for *randomized compilation* techniques to introduce diversity. Such diversity would preserve the functionality of well-behaved programs and be highly likely to disrupt others by removing unnecessary regularities.

****

Another group of researchers is working in University of Wales. The leaders of tjs grop are John E. Hunt and Denise Cooke Thei project is names ISYS (Generation and Visualization Mechanisms Based On The Immune System) The following is information from their web-site

Aims of the Project

The primary aim of this project is to enable the development of a toolkit for building learning systems based on the immune system. These systems should be capable of solving real-world problems in industry and commerce.

This involves further research into the algorithms, representations and operation of our Artificial Immune System. It requires the investigation of the capabilities (e.g., classification, prediction, control, etc.) of such a system on standard test suite problems. It should also consider the performance of the system on relevant real-world applications (which should be identified and investigated based on the results obtained from analysing the test suite problems).

We shall identify suitable test suites (e.g., from the Irvine repository of databases for machine learning research, University of California) by considering the amount of data available, how realistic the data is, and how representative it is of industrial and commercial problems.

It is important that we choose test problems which highlight the strengths of existing techniques as well as those which highlight the strengths of the AIS. This is necessary to obtain a fair comparison.

We shall do this using the extensive network of contacts available through the Centre for Intelligent Systems as well as through our collaborator's experience. We have already identified a number of potential industrial problem areas which we shall target once we have started the project.

Objectives of the Project

The main objectives of the research are listed below.

Development of the AIS theory. Our previous research has concentrated on developing the underlying concepts and algorithms of the AIS using an experimental approach. To develop the AIS further we need not only to consider the wider aspects of the immune system but also to formalise these theories and to understand the information processing operations being performed.

The implementation of a development toolkit. To promote the accessibility of the system to potential users we will develop a toolkit for building immune system based learning systems. This has proved an excellent way of enabling other approaches (e.g. neural networks and machine induction) to obtain both acceptance and significant financial benefits (e.g. ISL have found that the Clementine toolkit has been more acceptable to potential users due to its visual programming style interface).

The construction of a series of demonstrators which illustrate the performance of AIS based systems on test suite problems. These demonstrators will illustrate the capabilities of the AIS and allow comparison with existing techniques. We aim to show that it can out-perform existing approaches on certain classes of problem.

The construction of one or more systems to address real world problems. Although test suite problems are suitable for testing and comparison, it is still necessary to consider real world data. Therefore, we will illustrate the performance of the AIS on one or more real world problems. During the lifetime of the project we will be seeking to involve suitable companies from a range of industrial and commercial sectors.

The above objectives translate into a set of criteria against which the project should be assessed. That is, for the project to be a success we must show that we have a concrete theory (concepts and algorithms) on which the AIS is based and that we have produced a development toolkit which embodies this formalised theory. We must show that this tool kit can be used to construct a set of systems which address the test suite problems and that the AIS is useful for real world problems.

****
****

Jon Timmis (Department of Computer Science, University of Wales) participated in the development of the AIS by research group from Wales (including Hunt and Cooke). This research group extracted the primary principles of immunology to create the initial AIS for data analysis. Timmis has found out that the AIS through the process of cloning and mutation, built up a network of B cells that were a diverse representation of the data being analysed. This network was visualised via a specially developed tool. This allows the user to interact with the network and use the system for exploratory data analyses. Experiments were performed on two different data sets, a simple simulated data set and the Fisher Iris data set. Good results were obtained by the AIS on both sets, with the AIS being able to identify clusters known to exist within them. Extensive investigation into the algorithm's behaviour was undertaken and the way in which algorithm parameters effected performance and results was also examined.

Despite initial success from the original AIS, propblems were identified with the algorithm and the second stage of research was undertaken by the researches of the University of Wales. This resulted in the resource limited artificial immune system(RLAIS) which created a stable network of objects that did not deteriorate or loose patterns once discovered. In his atricle named "Artificial Immune Sustems: A novel data analysis technique inspired by the immune network theory" Jon Timmis presents a successful aplication of immune system metaphors to create a novel data analysis technique. He discribes principles of the RLAIS and claims that RLAIS goes a long way toward makin AIS a viable contender for effective data analysis.

****
****

Human ummune system, as it was told earlie, is massively distributed and parallel, highly adaptive and reactive, and maintains computational and biological reasoning functions which we are only just beginning to glimpse. Starlab(Belgium) is at the forefront of this technology, and has recently completed a fully functional artificial immune system test-bed, which is intended to be used both commercially and for academic research. To prove this new system's ability to learn and reason in real-time, a hardware monitoring experiment was conducted at Starlab's main laboratory.

The Starlab Artificial Immune System Shell is intended as a research test bed and investigative tool supporting Starlab's continuing development of expertise in the field. While continuing to undergo refinements, the core system (version 2.0) has been successfully validated with difficult machine learning sets and tested on real world data including cancer diagnosis sets, the risk assessment of malaria and anaemia cases as they occur in the field, and anomaly detection in real-world environments.

Modern physical security systems (like those which guard a building or outdoor site) generally have only two states: "on" and "off". Research group of the Starlab proved that AIS promises to enhance these (and many other) "dumb" systems with machine learning and enhanced observational capabilities or environmental "awareness". In order to test this approach in a practical setting, the Starlab AIS Shell was recently configured to accept real-time hardware input and to monitor external environments for anomalous behaviour.

The necessity for software spike filtering and signal normalising software was immediately apparent; and while some algorithms were designed and tested, they were not used to filter the streaming learning data as it was felt that the spikes and hardware abnormalities should also be learned by the AIS rather than simply being filtered out. For the monitoring phase a fuzzy matching system was needed to allow close-call combinations more leverage to match inline with the fairly high margin of hardware monitoring error (perhaps 2%!). This fuzzy system was built and tested with a variable threshold, although monitoring seemed to work best with around 90% fuzzy viability - well beyond the likely 2% margin of error; suggesting that the initial seeding of the antibody population was not sufficient (the initial gallery was of 10,000 randomly generated antibodies covering a possible spread of 4,398,046,511,104 combinations was reduced by about 40% following learning) and that genetic development of the antibodies (for maximum diagnostic coverage) would likely be a good solution to this bootstrapping problem of an over-narrow initial population. However, the fuzzy system here employed did prove the concept of "nearest neighbor" matching within an AIS.

Starlab developed " *Starlab Immune Networks* system ", it is a collection of algorithms and applications that model a distributed artificial immune system. It can be used to easily build complex applications that use the Artificial Immune System paradigm e.g. for intrusion detection, process control, design of secure software systems, physical security application, or distributed control applications.

The Immune Networks Engine is a collection of classes that implement the core algorithms needed in a distributed artificial immune system (DAIS). They also define the API for building DAIS applications. We built some applications that use this engine. One allows easy, graphical construction of AIS specification files used by the core classes as well as experimenting with the various AIS parameters. Another one is a distributed physical security application that can detect intrusions.

****

The project of Roger L. King, Aric B. Lambert, Samual H. Russ and Donna S. Reese (MSU/NSF Engineering Research Center for Computational Field Simulation, Mississippi State) is named " The Biological Basis of the Immune

System as a Model for Intelligent Agents". The primary objectives of this research are to gain an understanding of the recognition process, learning, and memory mechanisms of the immune system and how to make use of these functionalities in the design of intelligent agents. This research group claims that the primary functionalities of the human immune system to be captured in intelligent agents for an artificial immune system are:

- two step process of recognition followed by resolution (similar to content addressable memory)
- distributed responsibility among agents
- communication among agents
- evolutionary adaptation
- immunological memory
- self-organization
- judicious use of resources

Also, it has been proposed that two different types of agents be developed - H-cells for hardware management and S-cells for software.

H-cells and S-cells are envisioned as being designed as complementary intelligent agents. The S-cells have the role of defining the long term schedule of resources for execution of a program (i.e., a road map) and the H-cells provide near real-time control of the schedule (i.e., adjustments for the bumps and curves in the road). Where H-cells interrogate and monitor hardware resources during execution of a program for information, the S-cells interrogate the parallel program code for information prior to execution. This information is then used by the S-cells in planning how to schedule the code on the known distributed system hardware resources. The H-cells and S-cells are also distributed throughout the heterogeneous computing environment to focus on their assigned tasks, thus decentralizing the control process.

Initially, the S-cells may determine whether the code is data domain intensive or functional domain intensive. This determination can be made through either programmer supplied information or as a learned response to the program as it is executed. For example, in a data domain intensive code, the tasks to execute on all the hardware resources is functionally the same. However, the data may be different. In this case, knowledge about the nuances of the data (file size, precision requirements, etc.) will be important in assigning resources. In the case where the functionality of tasks differs (i.e., functional domain intensive), it is important to have
knowledge about the resource requirements of tasks (e.g., memory requirements, execution time, etc.). A simulation is a good example of a functionally intensive domain. Tasks will vary among the resources with different constraints for communication, execution, etc. The S-cell has the responsibility of determining critical paths for the simulation and allocating its distributed resource base in a judicious manner.

In the case of the H-cells, they are initially inoculated with basic knowledge about system hardware resources and behavioral properties. They will then have the tasks of learning about new hardware resources connected into the environment, monitoring resource usage, and managing system resources and processes. Inputs to the H-cells may include memory availability, CPU usage, disk traffic, and other performance related information. For the prototype S-cells it is envisioned that the software programmer will embed information about the code into pre-defined header slots. For example, information about critical paths may be specified, along with which tasks are computationally or communicationally intensive. Then, when a program is launched, the S-cells continue to monitor the activity of resources that they have planned to utilize along the execution path. If one of these resources becomes unavailable, it will be the responsibility of the H-cells to find an appropriate available resource in the short term. However, the S-cells will then reassess the road map and determine what resources it had planned to use to complete the tasks that may have to be remapped. After making any necessary adjustments to the road map, the new routes will be passed to the master unit for any appropriate action.

At present, this research group is on-going to develop intelligent agents to perform task allocation for a heterogeneous computing environment. Implementation of many of the functions of the human immune system described in this paper are being demonstrated in an H-cell implementation based on a hierarchical ART architecture. Data for inoculation of the H-cells and for defining the content addressable memory are being collected at both the system and task levels. At the system level, information is both static and dynamic. The static information includes: machine type, processor speed, operating system, internal memory available (both physical and virtual), page information, ticks per second, and number of processors. At the task level, all of the information is dynamic in nature. Task information includes: memory usage and a breakdown of a task ' s CPU usage into time for computing and communicating while executing an MPI task.

An intelligent control agent (ICA) based on the H-cell is being implemented to replace the slave agent used in the HECTOR system that helps globally manage the heterogeneous system. The artificial immune based scheduling system is designed around the decentralized decision-making of the ICAs. The ICAs perform all the decision-making functions of the HECTOR system. The ICAs are distributed among the nodes running parallel tasks, monitoring run-time performance and performing task control functions such as starting, migrating, restarting, and notifying tasks. System information such as memory, CPU, network, and disk usage are monitored to determent resource utilization of the node.

The ICAs run on all platforms that are candidates for executing parallel jobs. The primary focus of the ICA is performance optimization of the parallel tasks running on its node. Once every five seconds, the ICA takes performance measurements from the node it is residing on. A five-second interval was selected, because it provides a reasonable compromise between frequent updates and excessive system usage. Maintaining awareness of system performance is important for two reasons. First, it is the means by which the ICAs can detect external load and
determine the appropriate action. Second, it permits the agent to identify the machines that are most appropriate to run tasks. Because of the need of constant performance information a new capability has been added to the ICA.

This phase, known as the monitoring phase, gathers performance information, and structures it as a string of floating values for input into the intelligent controller portion of the agent.

The system calls used to gather the performance information were used from the original structure that was established for the slave agents used in HECTOR. On an unloaded single-processor Sun 110 MHz Sparc 5 workstation, the ICA uses 2.14 Mbytes of memory. Out of that, 617 Kbytes of it is resident. The measurements increased as the number of tasks increased, and the wall-clock indicated an additional 8.3 ms due to the extra tasks.

****

Researchers at Santa Fe Institiut with participation of Lee Segel (Applied Mathematics, Weizmann Insitute) and Stepahnie Forrest (Computer Science, University of New Mexico) are interested in the architecture and dynamics of the molecular networks that enable cells to perform complex, reliable decisionmaking under changing environmental conditions. They are also interested in the robustness and capacity for specific variation of these networks at the level of metabolic and genetic circuitry. Here the objective is to understand how the organization of control and regulatory mechanisms enables fast adaptation to changing environmental conditions, while preserving homeostasis. Secondarily, they plan to explore the range of viable alternative organizations and principles for control mechanisms.

A specific example of robust cell signalling networks are the cytokine signalling networks, which they propose to study in the context of the immune system (cytokine signaling systems are likely also crucial in other parts of the body as well). Each immune cell (e.g., a B-cell, T-cell, etc.) in the body (an estimated such cells are present at any given time) responds to a host of signalling molecules, called cytokines, which bind on the cell surface, stimulating a cascade of events within the cell. Likewise, every immune cell is capable of secreting a wide variety of cytokines under different conditions. At least 100 different types of cytokines (including the interleukins and interferons), are believed to participate in the immune response, but how they all work together has not yet been explained systematically.

What is known can be summarized (and greatly simplified) as follows. There are a large number of possible immune system (IS) effector mechanisms (e.g., mast cells, macrophages, and killer T-cells) each suited to different host tissue types and different pathogens. Innate (general) IS responses tend to be less specific and less efficient, but more quickly mobilized than adaptive (specific) IS responses. Molecular intercellular signalling via cytokines plays a large role in controlling individual effectors, affecting: differentiation of naive cell types into activated effectors, proliferation of effectors, deletion of ineffective or harmful effectors, recruitment of effectors to infected tissues or lymph organs, and activation of effector functions. Finally, each change in effector populations also changes the cytokine milieu, creating a complex feedback system. This aspect of the immune system is still poorly understood by immunologists and much of what is known comes from situations in which the system doesn't work properly--- animals that are susceptible to certain diseases, or knockout studies removing individual cell types or proteins, or from examining components of the system in isolation.

In spite of these impediments, a preliminary model of the complex network of signalling molecules and cells has been constructed by Lee Segel, a frequent SFI visitor and applied mathematician at the Weizmann Institute. Earlier attempts to model cytokine signalling pathways have concentrated on the effect of cytokines within a single cell. The Segel model concentrates on intercellular signalling pathways, specifically the network of interacting cells. Segel's model aspires to have the following properties: (1) A cell secretes multiple cytokines in response to a single stimulus; (2) Each cytokine is secreted by multiple cell types; (3) Each cytokine receptor is expressed on multiple cell types; (4) Each cell expresses multiple receptors, and there is crosstalk between intracellular signalling pathways, leading to amplification, inhibition, or mixing of signals; (5) The signals can be subverted (e.g., viruses can evolve to avoid or interfere with cytokines, e.g. by blocking receptors), so there is an evolutionary pressure towards robust, secure networks. In collaboration with SFI External Faculty member Stephanie Forrest, a computer scientist at the University of New Mexico, Segel plans to develop a more biologically plausible version of the model, compare it against known immunological data, and test various hypoetheses surrounding questions of robustness, including: How robust is the model to external perturbations? How big must the model be in order to exhibit robustness? How difficult is it for such models to evolve?

Initially, they plan to model the population dynamics of cells and cytokines, to identify different regimes of system behaviors, e.g., dominance of cellular or humoral immunity. They will do this using differential equations, ideas from statistical mechanics, genetic algorithms, and simulations of cell populations. Once they have some experience modeling simplified situations, they plan to move to modeling individual immune cells as small finite-state automata (FSAs), which communicate with one another via cytokines (possibly modeled as symbols or words). They can then study the global, or ``ensemble,'' properties of the collection of FSAs using techniques from statistical physics. They chose FSAs because they are readily applicable to many computational domains, so if they can build a reasonable cytokine model based on FSAs, then it would likely be applicable to many computational domains.

Cytokine signalling networks provide interesting clues about how to design a distributed autonomous control network which is dynamic (both the nodes and connections are changing in time), robust to small perturbations, but responsive to large perturbations (e.g., as would be important for large fleets of robots working together, for automated response in computer security, for mobile computing networks, or for other distributed intelligent systems).

This work was supported by the Commission of the European Communities

in the frame of the Project IST-2000-26016 IMCOMP.

Publications

1. Dipankar Dasgupta (Editor), *Artificial Immune Systems and Their Applications*, Publisher: Springer-Verlag, Inc. Berlin, January 1999.

2. Dipankar Dasgupta and Zbigniew Michalewicz (Editors), *Evolutionary Algorithms in Engineering Applications*, Published by Springer Verlag, Inc.Berlin, May 1997.

3. Dipankar Dasgupta (Section Editor) of a book *New Ideas in Optimization*, Publisher: McGraw-Hill Publishing, London, December 1999.

4. Dipankar Dasgupta and Fabio A. Gonzalez. Evolving Complex Fuzzy Classifier Rules Using a Linear Genetic Representation. Accepted for publication in the proceedings of the International Conference Genetic and Evolutionary Computation(GECCO), San Francisco, California, July 7-11, 2001.

5. Dipankar Dasgupta and Fabio A. Gonzalez. An Intelligent Decision Support System for Intrusion Detection and Response. To appear in Lecture Notes in Computer Science (publisher: Springer-Verlag) as the proceedings of International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security (MMM-ACNS), May 21-23, 2001, St. Petersburg, Russia. (Older version of the paper is available as CS Technical Report (No. CS-00-01), May 2000).

6. Dipankar Dasgupta and Hal Brian. Mobile Security Agents for Network Traffic Analysis. Accepted for publication by the IEEE Computer Society Press in the proceedings of DARPA Information Survivability Conference and Exposition II (DISCEX-II), June 12-14, 2001, Anaheim, California.

7. D. Dasgupta and F. Nino, 'A Comparison of Negative and Positive Selection Algorithms in Novel Pattern Detection', In the Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), Nashville, October 8-11, 2000.

8. D. Dasgupta, G. Hernandez and F. Nino, 'An Evolutionary Algorithm for Fractal Coding of Binary Image', In *IEEE Transaction on Evolutionary Computation, Vol 4, No. 2, July 2000.*

9. Immunology as information processing. S. Forrest and S.A. Hofmeyr. 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 (In Press). <ftp://ftp.cs.unm.edu/pub/forrest/iaip.ps>

10. Principles of a Computer Immune System. A. Somayaji, S. Hofmeyr, and S. Forrest. *1997 New Security Paradigms Workshop*, pp75-82, ACM (1998).

11. Computer immunology S. Forrest, S. Hofmeyr, and A. Somayaji. *Communications of the ACM* Vol. 40, No. 10, pp. 88-96 (1997).

12. "Automated Response Using System-Call Delays" A. Somayaji and S. Forrest. Usenix 2000.

13. Building diverse computer systems. S. Forrest, A. Somayaji, and D. Ackley. In *Proceedings of the Sixth Workshop on Hot Topics in Operating Systems*, Computer Society Press, Los Alamitos, CA, pp. 67-72 (1997).

14. Architecture for an Artificial Immune System. S. Hofmeyr and S. Forrest. *Evolutionary Computation Journal* (2000)

15. Immunity by Design: An Artificial Immune System. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, Morgan-Kaufmann, San Francisco, CA, pp. 1289-1296 (1999)

16. A sense of self for Unix processes. S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. In *Proceedings of 1996 IEEE Symposium on Computer Security and Privacy* (1996).

17. Intrusion Detection using Sequences of System Calls. S. A. Hofmeyr, A. Somayaji, and S. Forrest.
Note: this is the pre-submission version. A somewhat later version of this paper was published in the *Journal of Computer Security* *<http://www.iospress.nl/html/node449.html>* Vol. 6 (1998) pg 151-180 .

18. An Immunological Approach to Distributed Network Intrusion Detection. S. A. Hofmeyr, S. Forrest, and P. D'haeseleer. Paper presented at *RAID'98 - First International Workshop on the Recent Advances in Intrusion Detection* Louvain-la-Neuve, Belgium September 1998.

19. Detecting Intrusions Using System Calls: Alternative Data Models. C. Warrender, S. Forrest, B. Pearlmutter. *1999 IEEE Symposium on Security and Privacy* pp. 133-145 (1999).

20. A distributed approach to anomaly detection. P. D'haeseleer, S. Forrest, and P. Helman.
Submitted to *ACM Transactions on Information System Security* (1997)
Self-nonself discrimination in a computer. S. Forrest, A.S. Perelson, L. Allen, R. and Cherukuri. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, Los Alamitos, CA: IEEE Computer Society Press (1994).

21. An immunological approach to change detection: algorithms, analysis, and

implications. P. D'haeseleer, S. Forrest, and P. Helman. In *Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy* (1996).

22. An immunological approach to change detection: Theoretical Results. P. D'haeseleer. In *9th IEEE Computer Security Foundations Workshop* (1996).
23. Bersini, H. (1991) Immune network and adaptive control. *Proceedings of the First European Conference on Artificial Life*. (Ed. F. J. Varela and P. Bourgine). MIT Press.
24. Cooke, D.E., and J.E. Hunt (1995) Recognising Promoter Sequences Using An Artificial Immune System. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*. pp 89-97, Pub. AAAI Press, California.
25. Farmer, J.D., N.H., Packard, and A.S. Perelson (1986) The immune system, adaptation, and machine learning. *Physica D*, Vol. 22, 187-204.
26. Gilbert, C.J. and T.W. Routen (1994) Associative memory in an immune-based system. *Proceedings of AAAI'94*, AAAI Press, Menlo Park, California. Vol. 2, 852-857.
27. Hunt, J.E., and D.E. Cooke. Learning using an Artificial Immune System. to appear in the *Journal of Microcomputer Applications*, 1996.
28. J. E. Hunt, D. E. Cooke and H. Holstein, Case memory and retrieval Based on the Immune System, in the *First International Conference on Case Based Reasoning*, (October 1995) Published as Case-Based Reasoning Research and Development, Ed. Manuela Weloso and Agnar Aamodt, Lecture Notes in Artificial Intelligence 1010, pp 205 -216.
29. J. E. Hunt and D. E. Cooke, An Adaptive, Distributed Learning System, based on the Immune System, in Proc. of *the IEEE International Conference on Systems Man and Cybernetics*, pp 2494 - 2499, (1995).