An Artificial Immune Ecosystem Model for Hybrid Cloud Supervision

Fabio Guigou, Pierre Parrend, Pierre Collet

An Artificial Immune Ecosystem Model for Hybrid Cloud Supervision

Fabio Guigou[1,2,4], Pierre Parrend[1,3,4], and Pierre Collet[1,4]

[1] ICube Laboratory, Université de Strasbourg, France
pierre.collet@unistra.fr
[2] IPLine, Caluire-et-Cuire, France
fguigou@ipline.fr
[3] ECAM Strasbourg-Europe, Schiltigheim, France
pierre.parrend@ecam-strasbourg.eu
[4] Complex System Digital Campus (UNESCO Unitwin)
http://unitwin-cs.org/

Abstract. In this paper, we propose a new approach to the performance supervision of complex and heterogeneous infrastructures found in hybrid cloud networks, which typically consist of hundreds or thousands of interconnected servers and networking devices. This hardware and the quality of the interconnections is monitored by sampling specific metrics (such as bandwidth usage, CPU time, packet loss...) using probes, and raising alarms in case of an anomaly. We study an Artificial Immune Ecosystem model derived from the Artificial Immune Systems (AIS) algorithms to perform distributed analysis of the data collected throughout the network by these probes. In particular, we use the low variability of the measured data to derive statistical approaches to outlier detection, instead of the traditional stochastic antibody generation and selection method. The failure modes and baseline behaviour of the metrics being monitored (such as bandwidth usage, CPU time, packet loss...) are recorded in a distributed learning process and increase the system's ability to react quickly to suspiscious events. By matching the data with only a small number of failure signatures, we reduce the overall computations required to operate the system with respect to traditional AIS, therefore allowing its deployment on low-end monitoring servers or virtual machines. We demonstrate that a very small computational overhead allows the supervision engine to react much faster than the monitoring solutions currently in use.

Keywords: Artificial Immune Systems, Hybrid Cloud, Supervision

1 Introduction

In the recent years, Artificial Immune Systems have received some attention in applications related to data mining and clustering, such as document classification or intrusion

The work presented here has been founded by IPLine SAS, by the French ANRT in the frame of CIFRE contract 2015/0079, and by the French Banque Publique d'Investissement (BPI) under program FUI-AAP-19 in the frame of the HuMa project.

detection. Their distributed and dynamic nature as well as their ability to build up experience makes them highly suitable for anomaly detection in large datasets. However, their computational cost still limits their industrial adoption.

While the AIS model is no newcomer in the field of bio-inspired computing, it has not yet received nearly as much attention as other models such as evolutionary algorithms, neural networks or swarm computing [DMN07]. Its main drawback is the computational power required to perform classification and outlier detection [Hai11,SMTE05], typically by generating the antibodies and matching them with the antigens [FPAC94]. However, the increasing power of modern hardware has made them popular in some niche applications such as intrusion detection [ADG14] where their ability to adapt to a changing system, react more intensely to known patterns and work in distributed architectures was not found in other models.

In this paper, we present a new class of AIS-like bio-inspired frameworks, which we call Artificial Immune Ecosystems (AIE). We made up this term to refer to frameworks that retain the architectural and some conceptual elements of AIS, while mixing the strict biological model with other statistical tools. The works on software architecture related to computer immunology can be traced back to the late 90's [FPAC94,FHS97]. We then propose Service Level Monitoring of complex IT networks as a possible field of application for such a framework and demonstrate that a monitoring engine based on AIE, by memorizing earlier failures, can raise an alarm much faster than the traditional suites used in cloud supervision when a key metric (disk usage, CPU load, free memory, packet loss...) drifts away from its baseline behaviour. The investigation of possible immune responses to a detected anomaly is not in the scope of this paper.

The rest of the paper is organized as follows. In section 2, the state of the art of monitoring in the context of hybrid cloud, as well as the application of artificial immune systems as a framework for engineering robust IT ecosystems, are discussed. Section 3 presents our Artificial Immune Ecosystem model. Section 4 gives the experimental setup and results. Section 5 evaluates and discusses these results. Section 6 concludes this work.

2 Background

2.1 Monitoring

In the recent years, virtual infrastructures provided by cloud operators have begun to replace the traditional physical servers and network devices. Their success is mainly due to their scalability, cost-efficiency and simplicity when compared to datacenter hosting. Infrastructure-as-a-Service (IaaS) is the convergence of cloud computing, cloud storage and cloud networking: the end-user is provided with a complete virtualized ecosystem [VRMM11,MVML12]. Service providers no longer advertise network-only or VM-only platforms: they are now pushed towards an economic model in which they will have to operate core network infrastructure, system virtualization, network storage, WAN and Internet links and may even provide assistance in the LAN ecosystem. The latest model in IaaS is the hybrid cloud model: instead of keeping all his hardware on premises (private cloud) or fully externalizing it and accessing it through a public IP address (public cloud), the customer can mix those two paradigms, keep part of his computing local and reach delocalized resources through private or public addresses [SMLF09].

Service Level Monitoring (SLM) is a critical part of IT infrastructure management and a crucial point for modern cloud service providers [SMS+02,MJSCG04]. SLM is a prerequisite for enabling Service Level Agreements (SLA) [DWC10], *i.e.* the contractual level of infrastructure quality: availability, throughput, CPU, memory, etc. SLA define the risk level a customer is ready to accept. Current SLM solutions (Nagios, ntop, Sensu...) have two important shortcomings. On the one hand, are limited by their essentially centralized nature: while the probes themselves are distributed across the network, they are still managed by a central server storing the data and raising alarms in case of anomaly. Since the data is centralized in a single point, the IO and CPU load are very high, which requires powerful dedicated servers. On the other hand, they are stateless and only react to values crossing a given threshold or custom scripts raising an alarm. Though these systems keep the measured data for later analysis, it is not used as a knowledge base to predict future failures. They have no ability to learn from earlier failures, which makes root cause analysis and recovery complex and lengthy in a time-critical environment.

To overcome these limitations, shorten recovery times and allow for SLM scalability, we use a new artificial immune ecosystem model to propose a distributed, decentralized architecture that displays learning and knowledge sharing abilities. While other approaches [SC05] are very efficient at analyzing a signal or time series and detecting anomalies with respect to its history, they do not adress the problem of learning and recognizing the failure modes. AIS have also been used in this context [DF95], but with the traditional antigen/antibody model which requires the input to be converted to a bit string and generates a lot of detectors, and therefore computational overhead.

2.2 Artificial Immune Systems

The field of Artificial immune systems (AIS), also known as computer immunology [FHS97] emerged in the end of the 80's and the beginning of the 90's on the basis of the theory of idiotypic networks, which highlights the network-like connections between lymphocytes responsible for protecting the organisms against external threats [CBSV95]. First artificial models [FPAC94] draw on historical self vs. non-self selection in the natural immune system as conceptualised by Ehrlich [Sil05]. The second generation of models [HF99] lowers the detection threshold by taking co-stimulation, *i.e.* the simultaneous presence of several signals, into account [LC75]. The third generation of models [AC08] still refines the analysis by taking the impact of aggression into account: this is the danger model [Mat94].

The AIS domain has been a prolific field for theoretical endeavours [AHSGTV13], and knows a broad range of applications [ADG14]. However, its main focus so far is the application for data analysis and classification [TNH00] which proves to provide only limited performance [SMTE05,Hai11].

However, the mainstream AIS models are challenged by systemic models of natural immune systems, which characterizes core abilities of immunity such as memory, maturation, or ageing [TV14], as emergent properties of the network built by lymphocytes. The immune system would be not only a closed auto-regulated system for detecting aggressions and degradations, but a complete eco-system built from the very interactions between its components and with its environments. This shift in the analysis of immune systems provides huge perspectives for exploiting the adaptability and distributed

properties of immune systems for engineering eco-systems which are able to detect both known and de novo performance or security anomalies, and to perform suitable reactions in the context of evolving IT infrastructures.

3 Yet Another AIS model

One of the main drawbacks of AIS is the computational cost associated with the generation and maturation of the antibodies and the number of comparisons that must be performed for each incoming data point. While modern hardware is capable of handling such loads in experimental setups, we do not expect the amount of data acquired by a typical monitoring server (hundreds of probes, each yielding one to a few dozen measures per minute) to be analysed in real time on typical low-end servers. To overcome this limitation, we propose an alternative model in which only the previously recorded anomalies (failure modes) are recorded and matched against. Therefore only a handful of patterns are matched against and no random generation and maturation has to happen.

We combine the possibility to analyse new points with respect to the history of the system, the known failure modes and a predefined threshold by defining three different analysis modes which will be detailed later: traditional mode, lightweight mode and full mode. We explicit the properties of these modes in section 3.2.

3.1 Core features

While taking some distance from the bio-inspired model of the immune system, in particular to avoid the aforementioned performance drawbacks, the immune ecosystem we propose retains most of the AIS features, such as distribution, dynamic adaptation to a changing self and increased reaction after a secondary exposure, and puts aside the antigen/antibody model. It is based on the notions of innate immunity (failure modes already recorded by other probes), acquired immunity (failure detection by an alarm threshold, then pushed to other probes) and natural history (the past behaviour of the supervised system). Since the data related to earlier failures of the monitored devices is recorded, it would even be possible to transfer them from one instance of the AIE to another, which would represent a very basic vaccination process. However, the concept of antibodies and antigens, usually modelled in AIS by binary strings, is replaced by statistical analysis: instead, new data is matched against recorded signatures and a baseline, and the matching algorithm itself can be adapted to the problem.

Distribution Distribution is an often overlooked characteristic of immune systems [HVAF+14] that takes all its importance in the context of infrastructure monitoring. In biological immune systems, it allows for an immediate local response on the infection site. It is materialized by the omnipresence of antibody cells in the lymphatic system. In AIS, it enables local data storage and processing, hereby saving bandwidth and gaining overall efficiency. At the same time, a synchronized knowledge base must be available to all nodes at any time. It lists all the potential threats to the system.

Input and output The input to our system consists of real-valued time series. The immune ecosystem is triggered every time a new data point is added to the series (e.g. every minute in the context of a supervision system) or when a given number of points have been received (which may help with performance issues). It outputs alarm signals and updates to the global knowledge base.

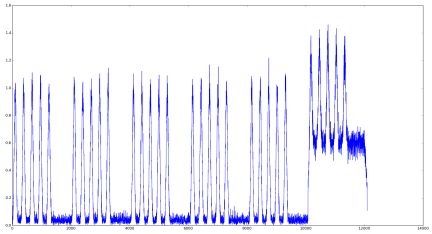

Fig. 1. Typical raw input data over six weeks

Figure 1 shows the data fed to the system over six weeks. This data shows the typical measures of the number of sessions on a firewall. The first five weeks are normal, with activity peaks on working days; week 6 shows an abnormally high number of sessions, which should be considered a failure.

The system also reacts to user input to classify the alarms and warnings in the knowledge base. This allows the operator to ignore mild transient threshold crossings as well as diagnose some abnormal behaviour as errors even though no threshold is crossed.

3.2 Architecture

We propose a fully distributed architecture in which a messaging bus is used to communicate between nodes and interact with the human expert. The probes capture data from the supervised systems (servers, networking devices, links...) and keep it locally for history and analysis. This creates a virtual database scattered across the nodes. At the same time, a shared knowledge base is cooperatively maintained by the probes. It keeps the signatures of all the failure events previously recorded. This knowledge base may be replicated for redundancy, up to the point where each nodes keeps a complete copy of it and can therefore operate even is case no other node is reachable. The messaging bus allows the probes to raise alarm signals and access and update the shared knowledge base. In the context of a distributed supervision system, it could also carry reconfiguration and deployment instructions for the probes.

Figure 2 shows the typical dataflow between the probes (P), devices, knowledge base (KB) and the expert. In our implementation, the probes, knowledge base and bus are run on the same servers, making the nodes homogeneous.

3.3 Operations

Overview Our AIE has three operating modes, each providing a finer analysis: traditional mode, lightweight mode and full mode. The traditional mode is simply the usual operation

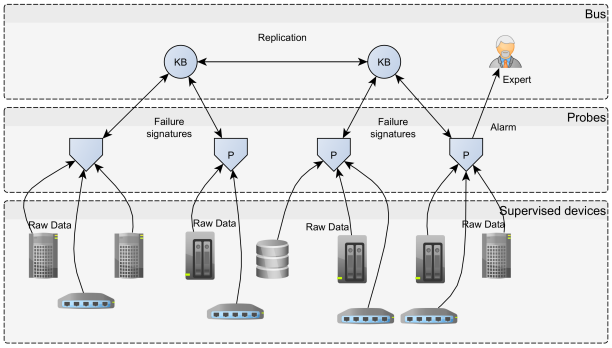

Fig. 2. Target architecture

of a regular SLM system, *à la* Nagios: each new data point is compared to a fixed threshold and, if that threshold is crossed, an alarm is raised. The lightweight mode adds a second check following the comparison: a window containing the k last data points is matched against the signatures of previous failures. Whenever a positive match is found, an alarm is raised, though with a smaller priority. It allows the system to detect the first signs of a potential failure before it actually happens. The full mode adds a third level of analysis by matching that same window against a long history (typically the last month). If no close enough match is found, then the window is considered an anomaly and a warning is raised. The human expert can choose to classify that event as a failure indication or ignore it. While this mode is by far the most computationally expensive, it is also the most powerful, since it may detect unknown failure modes before the incident takes place.

Details We first need to introduce a set of parameters for our AIE. Let n be the baseline length of the probe, *i.e.* the number of points kept locally as history, k the analysis window size, with $k << n$, m the signature length, a the sensitivity factor of the full mode and b the sensitivity factor of the lightweight mode.

In a typical monitoring setup, with one incoming point per minute, we may want to keep a baseline history of one month, as patterns often last one week (e.g. peaks on working days and flatlines during the week-end). A sound window size for anomaly detection is about two hours, which yields values of $n = 43200, k = 120$. $m - k$ is the time during which the early symptoms of an incident can be detected before the threshold is crossed, for which one hour is a reasonable choice, yielding $m = 180$. These values would be typical in a production but induce too many computations for experimental settings, therefore we worked with smaller values ($n = 8000, k = 24$ *and* $m = 36$).

In full mode, a history consisting of the last n data points is kept as the system baseline. Whenever a new point (or a given number of new points if computational cost is an issue) is received, a short window w of size k containing the last k samples is matched against the whole history w. A simple distance metric $d_i = \frac{1}{k}\sum_{j=0}^{k-1}|h_{i+j} - w_j|$ is computed at each position. We define $d_0 = min(d)$ and $i_0 = argmin(d)$, so that $d_{i_0} = d_0$. This allows the system to detect behaviour deviating from the baseline. If we define μ and σ as the mean and standard deviation of d_0 over the set of all newly

inserted points, we can define a threshold $\tau = \mu + a\sigma$ so that when $d_0 > \tau$ we raise an alarm indicating an abnormal behaviour. In experimental setups, we found $1 < a < 2$ to yield acceptable results; the results we present were obtained with $a = 2$. Figure 3 shows the influence of the choice of a, assuming a normal distribution of the measured distances. This configuration defines the notions of self and non-self for the AIE by detecting the relative position of a measure in the probability distribution.

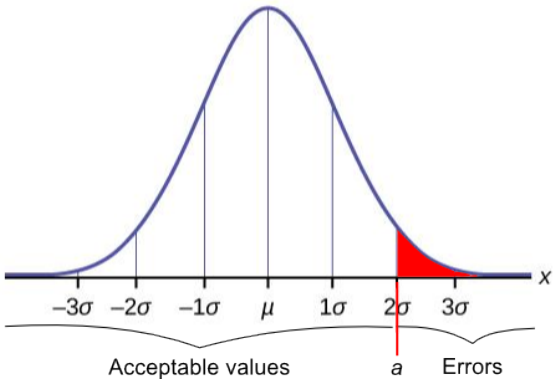

Fig. 3. Choice of a over a normal distribution

The lightweight mode avoids the computational cost of matching history. In this mode, the new data is matched against the known failure signatures (*i.e.* windows of m points that have been followed by a failure, with $m > k$) instead of the system baseline. If a positive match is found ($d_0 < b\delta_s$, where $\delta_s = \frac{max(s)-min(s)}{max(s)}$ models the variability of the recorded signature), an alarm is raised. We used a sensitivity factor of $b = 0.1$ after a trial-and-error search.

Lastly, when an alarm threshold is crossed, an alarm is raised as in any traditional monitoring system and the m points preceding the failure are added to the knowledge base. Note that in full mode, whenever an anomaly is detected, its signature is recorded and the expert can decide to tag it as a failure, in which case it is added to the signatures matched in lightweight mode.

Figure 4 shows the workflow for the execution in full mode. Exiting after the first box is equivalent to traditional threshold-based alarming, after the second means running in lightweight mode.

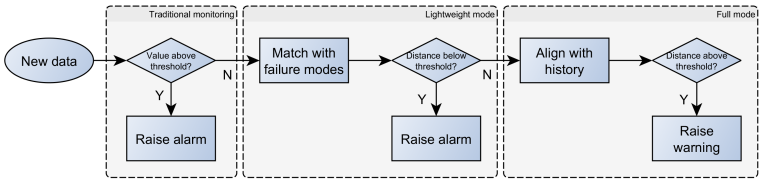

Fig. 4. Operating mode

4 First experiments

4.1 Setup

We built an experimental setup using a single probe which was fed with artificial data inspired from real supervision time series. A Redis database was setup to provide the communication channel and knowledge base. A control process listened to the PubSub bus to collect the alarm signals. We estimate the decoupling of these components sufficient to validate the architecture even with a single node, using multiple runs of one probe instead of parallel runs of a set of probes (the knowledge base being persisted between the runs). Figure 5 shows the data flow in the experimental setup. The data itself was generated using an alternation of bell curves and runs of zeros to which we added multiplicative and additive Gaussian noise, leading to the typical shape of CPU load plots.

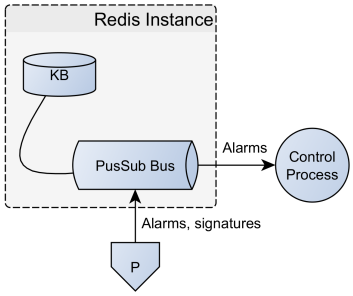

Fig. 5. Experimental setup

4.2 Results

We used the data presented in section 3. The system was run multiple times to train the lightweight mode. At the beginning of the failure event, figure 6 shows where the three stages send alarms. The failure signature is uploaded to the knowledge base, which allows the lightweight mode to react faster than the full mode after a primary exposure. The point corresponding to the lightweight mode was obtained after the first run, *i.e.* after the capture of a first incident window.

The experimental results show that the full mode, on an unknown failure, reacted 4 hours and half before the threshold was crossed, and the lightweight mode 42 minutes before on a known failure. The full mode was able to evaluate up to 5 points per second on a single-core process running on an Intel i5 CPU; the lightweight mode processed hundreds of points per second per signature. We therefore conclude that the lightweight mode may run on monitoring servers, but that the full mode would require optimizations and could only be used on servers monitoring only a handful of devices.

Figure 7 shows the reaction to a complete change of behaviour, which can be observed when monitoring the disk space used on a server. The space use begins to ramp up when a process begins to write a lot of logs. The full mode reacts at the very beginning of the incident, 3 days before the lightweight mode trained with the same failure.

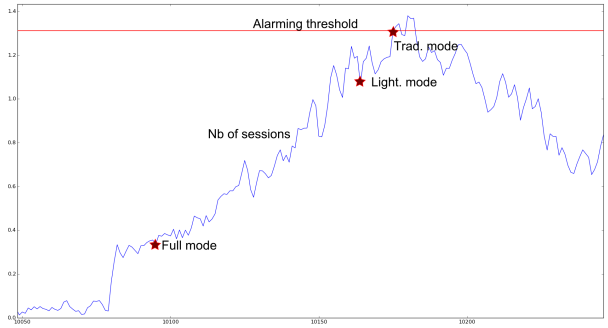

Fig. 6. Zoom on failure event, with the activation points and captured window

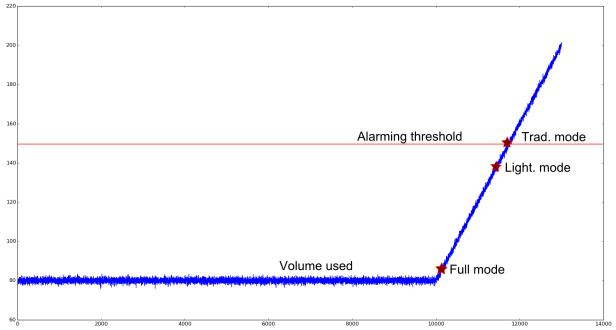

Fig. 7. Reaction to a complete change of behaviour

5 Discussion

5.1 Core properties of Artificial Immune Ecosystems

The Artificial Immune Ecosystems (AIE) we propose enables identifying weak signals through the multiplication of lymphocytes (abstracted as an event counter) and through the detection of signal dynamics far below traditional warning thresholds. It embeds the core structural properties of Artificial Immune Systems (AIS): distribution, input and output management, memory. Distribution is handled by the hierarchical structure of the AIE: local identification is performed through the lightweight mode, which performance overhead is compatible with the monitoring of typical servers; centralised detection is performed through the full mode which is more resource demanding and therefore limited to sensitive monitoring data and specific network nodes. Related to the input, only time series are supported so far, which already provides a great coverage for monitoring issues. Semantic data could be required for finer analysis in a second time. Related to the output, the ability of the current version of our AIE is to emit early warning to

enable quick reaction of a human operator. No automated or semi-automated reaction is supported so far. Memory is implemented through storage of known failure modes in a shared knowledge base.

Artificial Immune Ecosystems abstract and formalise the structural features of immune systems. However, existing benchmarks [SMTE05,Hai11] highlights the fact that data classification and outlier detection algorithms based on Artificial Immune Systems experience significant performance drawbacks with regards to statistical approaches like naive Bayesians. We are strongly convinced that a suitable abstraction of AIS, similarly to abstractions that exist in the domain of evolutionary algorithms, could provide a performant and adaptive solution for addressing evolving data as it appear in dynamic IT ecosystems.

5.2 Challenges

We have shown the ability of our specific AIE to detect unknown abnormal situations and known potential failures in scenarios where an error can be modelled by a threshold being crossed or an event not matching the past lifetime of the system. However, it is unable to raise alarms on more subtle events requiring correlation with other data sources or multiscale history matching. For instance, a high CPU load on a Sunday caused by an attack, if its shape is close enough to the normal load on a week day, would not give rise to any warning.

Moreover, we still need to experiment on our architecture in a real production environment where performance issues will arise and the actual data may display much more variability than our test data. We also need to study its behaviour on the long term, *i.e.* the growth rate of the knowledge base and the overall performance impact. Depending on these tests, we may introduce scaling functions in the signatures and a maximal lifetime for the failure modes. Our early tests have only validated the basic concepts; a mature software solution that could be deployed and used with any level of confidence by a cloud operator is yet to be designed and implemented.

Lastly, the only possible reaction of the system is to send a notification to a human expert, as with most SLM solutions. A more efficient way to react would be to activate more probes when a potential failure is detected [NS06,NS08,JKP+10], so as to collect more meaningful data, decide whether or not the alarm should be raised and help the expert with his analysis. Some control over supervised devices could also be handed to the AIE to allow some dynamic reconfiguration and prevent failures.

6 Conclusions and Perspectives

We have proposed a new computational and architectural framework inspired by the artificial immune systems and using statistical analysis where AIS tried to model antibodies and antigens: the Artificial Immune Ecosystem (AIE). To evaluate our proposal, we have implemented such an AIE for cloud infrastructure monitoring. By using multiple statistical tools, we made it possible to trade anticipation for performance and run only parts of the system where a large number of probes would have to be run in traditional AIS.

We still need to deploy this prototype in a real production environment to assess its actual efficiency and precision when confronted to real-world events. It will be up to the

end users to determine whether the configuration (sensitivity factors) is simple enough to be used on an industrial scale and if the errors (false alarms) in the first deployments are acceptable.

To conclude, we expect this new framework to open new perspectives to the field of artificial immunity by allowing experts to tune the comparison operator to the problem instead of tuning the problem to the AIS antibody/antigen model. We have already proved its interest in terms of architecture and modularity by building a distributed monitoring system with learning ability and acceptable performance. We plan to apply this framework and architectural implementation to other monitoring problems involving the distributed analysis of time series.

7 Scientific validation

This paper has been unanimously validated in a collaborative review mode with the following reviewers:

- Masanori Sugisaka (Alife Robotics, Japan)
- Juan-Julio Merelo (University Granada, Spain)
- Claudia Eckert (Open University, UK)

References

[AC08] Uwe Aickelin and Steve Cayzer. The danger theory and its application to artificial immune systems. *arXiv preprint arXiv:0801.3549*, 2008.

[ADG14] Uwe Aickelin, Dipankar Dasgupta, and Feng Gu. Artificial immune systems. In *Search Methodologies*, pages 187–211. Springer, 2014.

[AHSGTV13] Alaa Abi Haidar, Adrien Six, Jean-Gabriel Ganascia, and Veronique Thomas-Vaslin. The artificial immune systems domain: Identifying progress and main contributors using publication and co-authorship analyses. In *Advances in Artificial Life, ECAL*, volume 12, pages 1206–1217, 2013.

[CBSV95] Vera Calenbuhr, Hugues Bersini, John Stewart, and Francisco J Varela. Natural tolerance in a simple immune network. *Journal of theoretical biology*, 177(3):199–213, 1995.

[DF95] Dipankar Dasgupta and Stephanie Forrest. Novelty detection in time series data using ideas from immunology. In *In Proceedings of The International Conference on Intelligent Systems*, 1995.

[DMN07] Dipankar Dasgupta, N Majumdar, and F Nino. Artificial immune systems: A bibliography. *Computer Science Division, University of Memphis, Technical Report*, 2007.

[DWC10] Tharam Dillon, Chen Wu, and Elizabeth Chang. Cloud computing: issues and challenges. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 27–33. Ieee, 2010.

[FHS97] Stephanie Forrest, Steven A Hofmeyr, and Anil Somayaji. Computer immunology. *Communications of the ACM*, 40(10):88–96, 1997.

[FPAC94] Stephanie Forrest, Alan S Perelson, Lawrence Allen, and Rajesh Cherukuri. Self-nonself discrimination in a computer. In *null*, page 202. Ieee, 1994.

[Hai11] Alaa Abi Haidar. *An adaptive document classifier inspired by t-cell cross-regulation in the immune system.* PhD thesis, Citeseer, 2011.

[HF99] Steven Andrew Hofmeyr and Stephanie Forrest. An immunological model of distributed detection and its application to computer security. *The University of New Mexico*, 1999.

[HVAF+14] Farhoud Hosseinpour, Payam Vahdani Amoli, Fahimeh Farahnakian, Juha Plosila, and Timo Hämäläinen. Artificial immune system based intrusion detection: Innate immunity using an unsupervised learning approach. *International Journal of Digital Content Technology and its Applications(JDCTA)*, 8(5), 2014.

[JKP+10] D. Jeswani, N. Korde, D. Patil, M. Natu, and J. Augustine. Probe station selection algorithms for fault management in computer networks. In *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on*, pages 1–9, Jan 2010.

[LC75] Kevin John Lafferty and AJ Cunningham. A new analysis of allogeneic interactions. *Immunology and Cell Biology*, 53(1):27–42, 1975.

[Mat94] Polly Matzinger. Tolerance, danger, and the extended family. *Annual review of immunology*, 12(1):991–1045, 1994.

[MJSCG04] C. Molina-Jimenez, S. Shrivastava, J. Crowcroft, and P. Gevros. On the monitoring of contractual service level agreements. In *Electronic Contracting, 2004. Proceedings. First IEEE International Workshop on*, pages 1–8, July 2004.

[MVML12] Rafael Moreno-Vozmediano, Ruben S. Montero, and Ignacio M. Llorente. Iaas cloud architecture: From virtualized datacenters to federated cloud infrastructures. *Computer*, 45(12):65–72, 2012.

[NS06] M. Natu and A.S. Sethi. Active probing approach for fault localization in computer networks. In *End-to-End Monitoring Techniques and Services, 2006 4th IEEE/IFIP Workshop on*, pages 25–33, April 2006.

[NS08] M. Natu and A.S. Sethi. Application of adaptive probing for fault diagnosis in computer networks. In *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, pages 1055–1060, April 2008.

[SC05] Stan Salvador and Philip Chan. Learning states and rules for detecting anomalies in time series. *Applied Intelligence*, 23(3):241–255, December 2005.

[Sil05] Arthur M Silverstein. Paul ehrlich, archives and the history of immunology. *Nature immunology*, 6(7):639–639, 2005.

[SMLF09] B. Sotomayor, Ruben S. Montero, I.M. Llorente, and I. Foster. Virtual infrastructure management in private and hybrid clouds. *Internet Computing, IEEE*, 13(5):14–22, Sept 2009.

[SMS+02] Akhil Sahai, Vijay Machiraju, Mehmet Sayal, Aad van Moorsel, and Fabio Casati. Automated sla monitoring for web services. In Metin Feridun, Peter Kropf, and Gilbert Babin, editors, *Management Technologies for E-Commerce and E-Business Applications*, volume 2506 of *Lecture Notes in Computer Science*, pages 28–41. Springer Berlin Heidelberg, 2002.

[SMTE05] Thomas Stibor, Philipp Mohr, Jonathan Timmis, and Claudia Eckert. Is negative selection appropriate for anomaly detection? In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 321–328. ACM, 2005.

[TNH00] Jon Timmis, Mark Neal, and John Hunt. An artificial immune system for data analysis. *Biosystems*, 55(1):143–150, 2000.

[TV14] Véronique Thomas-Vaslin. A complex immunological idiotypic network for maintenance of tolerance. *Frontiers in immunology*, 5, 2014.

[VRMM11] Luis M. Vaquero, Luis Rodero-Merino, and Daniel Morán. Locking the sky: a survey on iaas cloud security. *Computing*, 91(1):93–118, 2011.