Soft Computing Techniques for Internet Backbone Traffic Anomaly Detection

Antonia Azzini[1], Matteo De Felice[2,3], Sandro Meloni[3], and Andrea G.B. Tettamanzi[1]

[1] University of Milan, Information Technology Department
{azzini,tettamanzi}@dti.unimi.it
[2] ENEA (Italian Energy New Technology and Environment Agency)
[3] University of Rome "Roma Tre", Department of Informatics and Automation
{defelice,sandro}@dia.uniroma3.it

Abstract. The detection of anomalies and faults is a fundamental task for different fields, especially in real cases like LAN networks and the Internet. We present an experimental study of anomaly detection on a simulated Internet backbone network based on neural networks, particle swarms, and artificial immune systems.

1 Introduction and Problem Definition

In computer networks, anomaly detection is currently one of the hottest topics, whose unambiguous goal is searching for potential security breaches. The study of techniques for the detection and prevention of traffic anomalies has spanned a rich spectrum of paradigms. Nonetheless, it still represents a difficult challenge, not only because the Internet infrastructure is not designed to detect anomalies, but also because volume anomalies can take a wide range of different forms, each in principle characterized, at the onset, by a different traffic profile.

The problem of anomaly detection can be naturally recast into a classification problem. In such a scenario, several approaches have been presented in the literature, with an increasing interest in machine learning and in the application of nature-inspired algorithms, such as evolutionary algorithms (EA), artificial neural networks (ANN), and artificial immune systems (AIS). This work discusses some possible applications of such techniques to anomaly detection, showing the most important aspects of the considered methodologies applied to two examples of fault cases.

One of the most typical examples of anomaly, related to computer security, regards Network Intrusion, an attack to a system coming through the network where computers communicate via standard protocols. Other typical examples of anomalies are traffic volume anomalies, generated, for example, by denial-of-service (DoS) attacks or flash crowds. They consist in a large surge in traffic to a particular site causing a dramatic increase in server load and putting severe strain on the network links leading to the server, which, in turn, results in a considerable increase of packet loss and congestion, degrading network operation and impacting the perceived quality of service (QoS) for the user [11]. It is therefore important to be able to timely detect them from the onset.

M. Giacobini et al. (Eds.): EvoWorkshops 2009, LNCS 5484, pp. 99–104, 2009.

The remainder of this paper is organized as follows. Section 2 describes the detector models and the algorithms implemented in this work, while section 3 presents the experiments carried out and discusses the results obtained. Finally, Section 4 provides some concluding remarks.

2 Detector Models and Detector Set Generation

In the anomaly detection problem, different kinds of models are used to define detectors. Some make use of ANNs [10], while others consider geometric coverages by using, for example, hyperspheres, hypercubes, or hyperellipses [1,3]. In this work, we consider three types of detectors: two kinds of ANNs, respectively feed forward (FFNs) and radial basis functions networks (RBFNs), and hypersphere sets:

- *Hyperspherical Detectors* consisting of a set of coordinates, representing a point in the n-dimensional feature space, and a radius value (hyperradius).
- *Artificial Neural Networks*, able to distinguish between normal and anomalous pattern after observing a set of pattern during a learning phase (*training phase*). The classification obtained is an index that varies from `normal` to `anomalous`, which we mapped respectively to the real values 0 and 1. In this work Feed-Forward Networks (FFNs) and Radial Basis Function Networks (RBFNs) are considered, described in detail in [7].

Three different approaches are implemented for optimal detector definition: the usual error backpropagation algorithm for training FFNs, Particle Swarm Optimization (PSO) for optimizing hyperspherical detectors, and negative selection (NS) of Artificial Immune Systems. The main features are reported below, while the specific parameter settings used in this work are reported in Table 1.

Particle Swarm Optimization is a population-based optimization technique [2,12], whereby each individual moves in the solution space following two main attraction points: the best solution it has encountered so-far and the best solution found by any other individual in the swarm at a given time.

Only normal cases are considered and the fitness function is defined as the ratio between the true positive range of normal cases for each detector (also indicating the density of each detector) and its area, penalizing possible overlaps between detectors in the positive recognition.

A modified version of the traditional PSO is implemented in this work in order to speed up convergence. Low-fitness detectors (i.e., low density of normal traffic detected) are attracted to the best ones, while the worst detectors, with no normal coverage, are reinitialized. Furthermore, detector diversity is maintained through a *repulsion* mechanism, in order to reduce their overlaps. Finally, at each iteration, while the fitness does not worsen, the radius of each detector is reduced (*radius self-tuning*). Such modified PSO allows to avoid hypersphere detectors with too large radii, thus improving the coverage of the normal space and reducing holes.

Neural Networks Training: NNs are used for the non-anomalous space covering. Generally, for real problems, the set of positive (non-anomalous) istances

is smaller than the complementary set of anomalous instances. For this reason, in this work we train ANNs both with the positive set and some randomly-generated noise patterns, as fault cases.

Negative Selection: is one of the major algorithms developed in the field of AIS [8]. It generates a set of detectors covering the complementary space to the normal one, in order to classify new, unseen data as normal or anomalous. The detector set is defined through randomly generated detectors that do not recognize normal samples as anomalous.

3 Experiments

In order to compare the behavior of the considered detector models, all the experiments carried out in this work refer to anomalous events in a heavy-traffic network, the Abilene Internet 2 Backbone[1], by observing signals on a subset of the network's nodes and links. A network simulator is used to generate the dataset of the network traffic signals.

The Abilene Internet 2 network is a US continental backbone, connecting several educational, research, and commercial institutions. The main backbone consists of 9 macro-nodes connecting some of the main US cities and 14 OC-192 (10-gigabit ethernet optical fiber cable) links. The average traffic volume is about 1 Gb/s over the links, with two different kinds of periodical oscillations, representing respectively daily (day/night) and weekly fluctuations (week end/working days). In the Abilene network modeling, efforts have been spent to reproduce the typical traffic volumes and shapes over the links starting from an Origin/Destination Matrix (OD-Matrix). Although this is known as a generally NP-hard problem, an approximate solution can be obtained by studying the topology and the behavior of the network.

The software used to define the network is Network Simulator 2 (NS2)[2], that reproduces all the traffic features as the TCP/IP protocol stack, and it is able to generate traffic information qualitatively and quantitatively comparable with the actual Abilene backbone. Two different link fault cases have been created with this simulator for the experiments.

Table 1 shows the most important parameters considered for each model, together with the values that they take up in the experiments carried out. All the detector models are tested over 10 runs for each parameter setting. The bold value for each model corresponds to the best setting.

3.1 Results and Discussion

Two groups of experiments are carried out in this work, by considering, respectively, the flow through the nodes and through the links of the simulated network. The results of the latter group of experiments are shown in detail, since they

[1] http://www.internet2.edu/network/
[2] http://www.isi.edu/nsnam/ns/

Table 1. Parameter Settings for the detector models: FFNs with the backpropagation algorithm (`BP-FFNs`), RBFNs with the training algorithm (`RBFNs`), Negative Selection with FFNs (`NS-FFNs`), Negative Selection with RBFNs (`NS-RBFNs`), Negative Selection with Hyperspherical Detectors (`NS-HDs`), PSO with Hyperspherical Detectors (`PSO-HDs`). The bold values for each model correspond to the best settings.

ANNs	`BP-FFNs`	Learning Rate: **Adaptive**; Training: **Resilient Backpropagation**; Transfer functions: **[logsig - purelin]**, [tansig - purelin]; Network topology: **[3-3-1]**; Stopping criterion: max epochs (1000), minimum gradient ($1e^{-10}$)
	`RBFNs`	Training: **MATLAB** `newrb`; Maximum number of neurons: 20,30,**50**
PSO	`PSO-HDs`	Maximum number of iterations: 200,**500**; Number of particles: 10, **25**; Number of detectors modeled by each particle: 5, **8**; Radius Self-Tuning: **yes**, no; Repulsion: **yes**,no
NS	`NS-FFNs`	Number of detectors: **10**; Transfer functions: **[logsig - purelin]**, [tansig - purelin]; Network topology: **[3-3-1]**; Stopping criterion: max epochs (1000) / minimum gradient ($1e^{-10}$); Acceptance threshold: **0.2**
	`NS-RBFNs`	Number of detectors: **10**; Spread range: [100,700], **[300,700]**; Detectors radius range: **[10,250]**, [10,300]; Acceptance threshold: **0.2**
	`NS-HDs`	Number of detectors: **100**, 150; Detectors Radius: **variable**, costant

present more interesting and more effective results over the considered traffic network. In the link-flows, the variations are less dramatic after the link fault, especially if we use links not strictly involved with the fault as inputs. Figure 1 shows the results obtained from the simulated fault case of the link between the cities of Chicago and Atlanta.

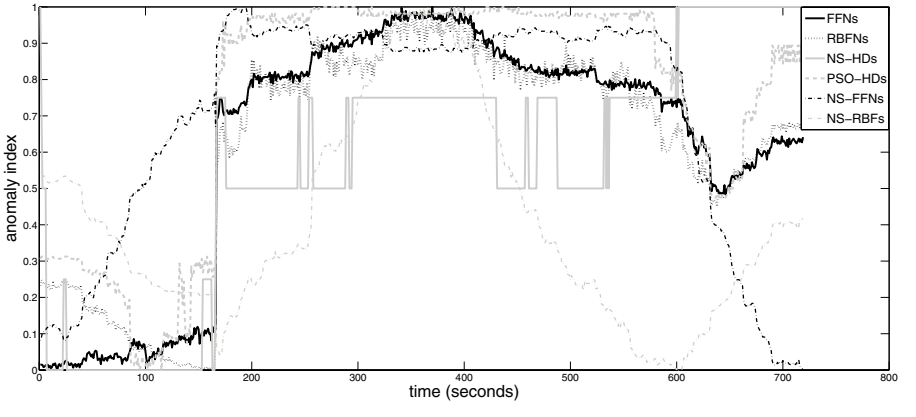

Fig. 1. Comparison of the best results obtained from the detector models

In this case, after the link fault, the traffic routing changes and does not revert to the previous situation even after the links come back up. Indeed, we can see in Figure 1 that after the fault around second 160, the anomaly index remains high for a very long period, even if the fault ends at time 250. The only technique which does not detect the fault properly is `NS-RBFs`, while the others show a visible edge around the fault event, although the edge of `NS-FFNs` is smooth and small.

ANNs show good performances on all the experiments conducted, especially `FFNs`, representing a really common and well-studied tool for a wide variety of applications. Nevertheless, the fact should be emphasized that ANN training requires a complete dataset with istances of both classification classes, i.e., normal and anomalous data, which usually is not avalaible, because a complete characterization of all the possible anomalies is not available in general.

We consider the performance of PSO (`PSO-HDs`) really encouraging: our modified version of the PSO algorithm is able to detect almost all of the faults presented, even when considering an offline anomaly detection problem only. Indeed, the nature of this algorithm makes it a convenient choice for real-time detection systems and well-suited for non-stationary environment. The main advantage of this technique is the easiness of implementation and extension with new functions, as well as its fast execution time with any coding language.

NS-based methods are usually a good choice for anomaly detection problems. We observe in our experiments good performances both with hyperspherical detectors and with ANNs, even if the latter need a more accurate parameters tuning, a characteristic often observed with ANNs.

The models considered in this work have been widely applied in different approaches presented in the literature regarding the anomaly detection problem. One of the first observations carried out in the survey by Kim and colleagues [13] is that various features of the NS algorithm make it by far the most popular algorithm for solving such problems. However, despite its appealing properties, it has not shown great success in real-life applications. The authors indicate two drawbacks to utilizing the NS algorithm, namely scalability and coverage, defining them the main barriers to their success as an effective detection model.

Timmis and Freitas [9] too indicate the use of a NS-based AIS as problematic, presenting, besides those indicated in [13], other disadvantages; indeed, the random generation of detectors is not adaptive and does not use any information to guide search. Then, the lack of mechanisms to minimize overfitting, and the fact that NS is mainly an algorithmic rather than a problem-oriented approach, produce poor solutions in the model definition.

In the literature, analogies with other intelligent techniques were and are still today appropriate ideas in order to improve NS in anomaly detection problems, with particular attention to evolutionary approaches, like genetic algorithms and PSO, artificial neural networks and, in some cases, also fuzzy rules. Furthermore, joint negative and positive selection could be a satisfactory solution. Hybrid representations that use evolutionary approaches in analogy with AIS could become useful in order to achieve a high level of robustness and adaptability. Moreover, solutions implemented with neural networks require a reduced number of detectors *vis à vis* those that use geometrical detectors, also by considering simple topologies, thus reducing the overall computational cost.

4 Conclusion and Future Work

In this work, a particular attention has been paid to applications based on nature-inspired algorithms for a normal vs. anomalous network traffic classification.

Different detector models have been applied, together with different detection algorithms, to two simple cases of fault detection. The results obtained from the experiments carried out show how all such techniques may obtain satisfactory results even without an in-depth preliminary analysis and tuning of the parameters for each implemented algorithm.

Future works will consider more difficult network traffic situations, with more irregular traffic data (by considering, for example, LAN traffic information), or real-time fault detection. Moreover, other algorithms should be considered for network traffic analysis, for example CLONALG [5] and Immune Networks [6].

References

1. Balachandran, S., Dasgupta, D., Nino, F., Garrett, D.: A framework for evolving multi-shaped detectors in negative selection. In: Proc. of IEEE Symposium on Foundations of Computational Intelligence, FOCI 2007, pp. 401–408 (2007)
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: From natural to artificial systems. Oxford University Press, Oxford (1999)
3. Bouvry, P., Seredynsky, F.: Anomaly detection in TCP/IP networks using immune systems paradigm. Computer Comm. 30, 740–749 (2007)
4. Dasgupta, D., Ji, Z.: Real-valued negative selection algorithm with variable-sized detectors. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 287–298. Springer, Heidelberg (2004)
5. De Castro, N.L., von Zuben, F.J.: Learning and optimization using the clonal selection principle. IEEE Trans. on Evolutionary Computation 6(3), 239–251 (2002)
6. De Castro, N.L., von Zuben, F.J.: Immune and neural network models: Theoretical and empirical comparisons. Int. Journal on Computational Intelligent Applications 1(3), 239–257 (2001)
7. Dreyfus, G.: Neural networks, methodology and applications. Springer, Heidelberg (2005)
8. Forrest, S., Perelson, A., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: Proc. of the IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA, pp. 202–212 (1994)
9. Freitas, A.A., Timmis, J.: Revisiting the Foundations of Artificial Immune Sustems for Data Mining. Trans. on Evolutionary Computation 11(4) (August 2007)
10. Gao, X.Z., Ovaska, S.J., Wang, X., Chow, M.Y.: A neural networks-based negative selection algorithm in fault diagnosis. Neural Computing & Applications 17, 91–98 (2007)
11. Jung, J., Krishnamurthy, B., Rabinovich, M.: Flash crowds and denial of service attacks: Characterization and implications for cdns and web sites. In: Proc. of. WWW 2002, Hawaii (May 2002)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. of IEEE International Conf. on Neural Networks, Perth, Australia, vol. 4, pp. 1942–1948 (1995)
13. Kim, J., Bentley, P.J., Aickelin, U., Greensmith, J., Tedesco, G., Twycross, J.: Immune system approaches to intrusion detection - a review. Natural Computing 6, 413–466 (2007)