USING FUZZY SYSTEM TO MANAGE FALSE ALARMS IN INTRUSION DETECTION

Mehdi Shajari*
National Research Council
46 Dineen Drive
Fredericton, NB, E3B 9W4, Canada
Mehdi.Shajari@nrc.ca

Ali A. Ghorbani
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, E3B 5A3, Canada
ghorbani@unb.ca

Abstract The Fuzzy Adaptive Survivability Tools (FAST) is an intelligent multi-agent based intrusion detection system that survives the network in the face of large scale intrusion problems. The proposed system is based on automated detection and response approach for survivability. It identifies anomalous host and system variables and uses them to detect known attacks and events of interest. The system uses different intelligent agents to identify normal and abnormal patterns automatically and adaptively. Fuzzy logic is used to discover the underlaying structure of normal and misuse patterns. The simulation results obtained with KDD CUP 1999 data set indicates that the proposed system can effectively manage false alarms.

Keywords: Intrusion Detection, intelligent agent, false positive, false negative, fuzzy system, network survivability.

Introduction

One of the biggest problems facing Intrusion Detection Systems (IDS) is the number of false positives. Each false alert has a cost associated with it. Moreover, it keeps the security operation from focusing on the

*The first author is a Ph.D. student at the Faculty of Computer Science, University of New Brunswick, Fredericton, NB, E3B 5A3, Canada.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35691-4_52
D. Gritzalis et al. (eds.), *Security and Privacy in the Age of Uncertainty*

true alerts. The main difficulty is the fact that it is not always easy to tell the difference between a true and a false alarm.

There are two types of false positives. The first type of false positives are generated when an attack detection algorithm misclassifies normal traffic as an attack. This is normally due to a loophole in the detection algorithm. The second type of false positives (referred to as false alarms in (ISS01)) are generated when a detection algorithm correctly identifies a given pattern, but the pattern does not represent a true threat. This is mainly due to misconfiguration and constitutes the majority of false positives generated by an intrusion detection system. The second type of false positives is usually dealt with at the client sites where security operators configure IDS to reduce false alarms. In this paper we refer to both types of false positives simply as false positives. The term *false alarm* refers to all kinds of false alerts including false positives and false negatives.

Computer systems and networks continue to grow in size and complexity and even top experts are too slow to analyze the problems and react to them. This is simply impossible. It is therefore desirable to design a system that can simulate the experts knowledge and quickly react to the attacks. Fuzzy logic can best be applied to problems that rely heavily on human experience and intuition.

Currently, we are developing a multiagent-based intrusion detection and response system called Fuzzy Adaptive Survivability Tool (FAST). The system is based on automated detection and response approach for survivability. It identifies anomalous host and system variables and uses them to detect known attacks and events of interest. FAST consists of a knowledge-base and five different types of autonomous agents, namely: monitor agent, detection agent, decision agent, action agent and interface agent. The knowledge-base part of the system includes top security expert's knowledge, common sense reasoning and reactions to attack. This paper present the management of false alarms by FAST.

The remainder of this paper is organized as follows. Section 1 describes fuzzy partitioning of host and network variables. Design components are introduced in Section 2. The experimental results of a prototype implementation of FAST are presented in Section 3. Finally, the conclusions of the present study are summarized in Section 4.

1. Fuzzy partitioning of host and network variables

Fuzzy sets and probability theory are two most powerful tools to overcome uncertainty. Fuzzy membership functions represent similarities of

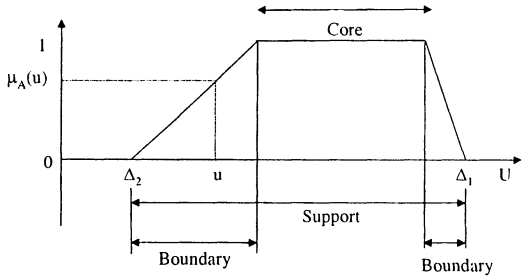

Figure 1. Core, support and boundaries of a fuzzy set, adapted from (Ross00).

objects to imprecisely defined properties, while probabilities convey information about relative frequencies (Dubois93). False alarms are the results of uncertainties in the detection of attacks. The uncertainties are mainly due to the inherent fuzziness of the host and networks variables. We propose to use fuzzy set theory as a mathematical framework for managing false alarms.

The core of a membership function for a fuzzy set A, $core(A)$, is defined as those elements of the universe, u, where $\mu_A(u) = 1$. The support of a membership function for a fuzzy set A, $supp(A)$, is defined as $0 < \mu_A(u) \leq 1$. The boundaries of a membership function is decided by $0 < \mu_A(u) < 1$. Figure 1 illustrates the regions for the core, support and boundaries of a typical fuzzy set.

Building membership functions is the most important part of the fuzzy modelling. The first method that may come to mind especially for the anomaly detection is converting frequency histograms or probability curves to membership functions. However, it should be remembered that the membership functions are not probabilities. In FAST, monitor agent monitors network variables and builds a fuzzy partition for each variable. Let $Q = \{A_1, .., A_n\}$ be a family of fuzzy sets on U. Q is a fuzzy partition of U when (Dumitrescu93)

$$\sum_{i=1}^{n} A_i(u) = 1, \quad u \in U. \tag{1}$$

Monitor agent's primary job is to find the $core(normal)$ and its boundaries in order to build normal membership function. *normal* is a membership function that describes the normal behavior of a variable. Monitor agent uses different definitions for building membership functions. The following are defined for each variable:

1 Number of membership functions: 2 partitions (*normal* and *abnormally-high*) or 3 partitions (*normal* and *abnormally-low* and *abnormally-high*).

2 A method for avoiding the anomalous observations in the training, so that the monitor agent can accurately learn the normal behavior of the variable.

3 A method for determining core of a normal membership function.

As part of their observations, monitor agents collect five different statistics (without considering the frequencies): the minimum value, the first quartile, the median, the third quartile, and the maximum value (see box plot in Figure 2). Any of these values as well as the range between any two values can be selected as $core(normal)$.

To build normal membership function, monitor agent starts with values, x's, that are normal (i.e., $\mu_{normal}(x) = 1$). Experts commonly choose median of the monitored values as the best representative of a normal situation. In such case, we have $\mu_{normal}(median) = 1$. Minimum, first or third quartiles values are also possible choices. Any range between any two of these values is also a reasonable choice in different situations. To avoid false positives, monitor agent assigns 0.5 as the degree of membership to the normal partition for the highest and lowest observed values that are believed to belong to normal operation (i.e., $\mu_{normal}(min) = 0.5$ and $\mu_{normal}(max) = 0.5$). Then, the positions of Δ_1 and Δ_1 (see Figure 2) can be decided to satisfy this assignment.

Figures 2(a) and 2(c) show normal membership functions for variables with three and two fuzzy numbers, respectively. Figures 2(b) and 2(d) show fuzzy partitions that are built based on Equation 1. When a variable is partitioned to *normal* and *abnormally-high*, only one of the values from five-number summary will be selected. Figure 2(d) shows the membership functions of the fuzzy set pair *normal* and *abnormally-high* and $normal(u) + abnormal(u) = 1$ for any u of observed values.

2. Design Components

2.1 Rule-base

A set of fuzzy rules representing the security experts' knowledge is defined. The rules are represented as

```
IF Condition (Premise) THEN Action (Conclusion)
```

In each rule, the condition and the conclusion consist of the statements in the form of:

```
(Variable) is (Term)
```

and are connected together by the standard logic operators. The representation of knowledge using fuzzy sets allows us: a) to apply a rule

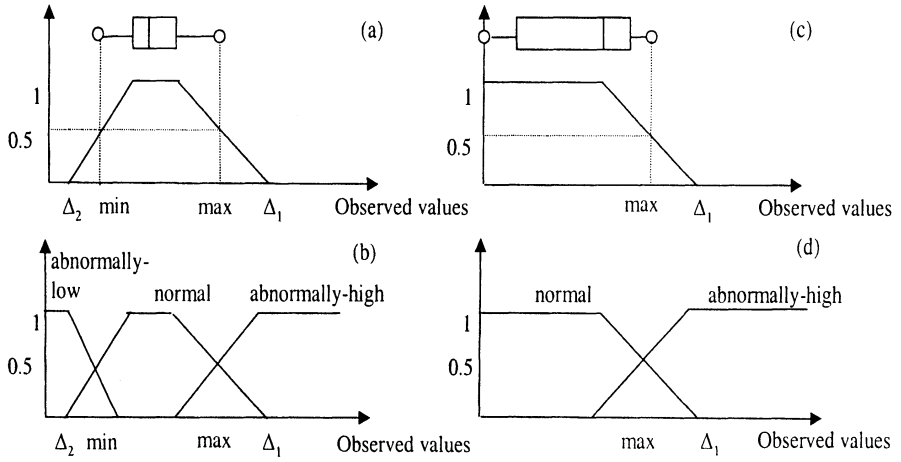

Figure 2. (a) and (c) Building normal membership functions, (b) and (d) fuzzy partitioning of a variable (note that the 5-number summary is shown by the box-and-whiskers plot).

even when the data does not match exactly the premise of the rule, b) to avoid the difficulty of determining the strict intervals for each linguistic term and c) to prevent the increase in the number of rules due to interval division. Moreover, a fuzzy rule system provides an interpolation mechanism for the different knowledge introduced. Thus, with a minimum number of rules, the system is able to respond to incomplete situations (Jaulent00).

In FAST, each detection agent is responsible for detecting an event of interest. This event can be very general, like probing in general or a specific event such as probing by one of the known probing tools. Table 1 shows examples of rule sets for detection agents.

2.2 Membership Functions

The shape of any particular membership function is often depends on the application. In FAST, the membership functions can be of linear, triangular or trapezoidal shapes. Figure 3 shows the membership functions for *same-host-diff-srv-in-time-window* variable. Figure 3(M1) illustrates the case where an agent uses the minimum value. Figure 3(M2) shows the membership functions where monitor agent uses the first quartile as the normal range for the variable. Figure 3(M3) illustrates membership functions when median of the observations indicates the normal range for the variable.

Each detection agent uses fuzzy inference to categorize a record in one of the three categories (no similarity to an attack, suspicious to be

Table 1. Examples of Rule sets for detection agents.

Type of Attack	*Rules in the rule-base of detection agent*
General Probing Attack	**Rule 1:** IF (same-host-diff-srv-in-time-window IS normal) AND (same-srv-diff-host-in-time-window IS normal) AND (same-host-diff-srv-in-recent-connections-window IS normal) AND (same-srv-diff-host-in-recent-connections-window IS normal) THEN probing IS not-detected
	Rule 2: IF same-host-diff-srv-in-time-window IS abnormally-high THEN probing IS certain
	Rule 3: IF same-srv-diff-host-in-time-window IS abnormally-high THEN probing IS certain
	Rule 4: IF same-host-diff-srv-in-recent-connections-window IS abnormally-high THEN probing IS certain
	Rule 5: IF same-srv-diff-host-in-recent-connections-window IS abnormally-high THEN probing IS certain
Neptune (Synflood)	**Rule 1:** IF S0 AND same-host-syn-error-in-time-window IS normal THEN neptune IS not-detected
	Rule 2: IF S0 AND same-host-syn-error-in-time-window IS abnormally-high THEN neptune IS certain
Smurf	**Rule 1:** IF ecr-i AND count IS normal AND srv-count IS normal THEN smurf IS not-detected
	Rule 2: IF ecr-i AND count IS abnormally-high AND srv-count IS abnormally-high THEN smurf IS certain

an attack, similar to an attack). For example, probe detection agent examines each record to see if it is similar to a probing attack or not. It may then label the record as a suspicious probing attack. Neptune detection agent does the same to see if the record belongs to neptune attack or not. A record may be labelled as both probing and neptune by two different agents.

The shapes of the output functions for a detection agent is configurable. Figure 4 shows different output functions used in our experiments. FAST can manage and control the rate of false negatives by tuning the output functions. The number of false negatives is decreased by increasing the range of suspicious membership (see Figure 4(O1)). At the same time, this output function generates more false suspicious alarms. As we narrow down the suspicious membership function (see Figure 4(O2-O5)), the number of false negatives is increased, but less false suspicious alarms are generated. Suspicious function can further be divided to demonstrate more levels of suspicion. Figure 4(O6) shows an output function with three suspicion levels.

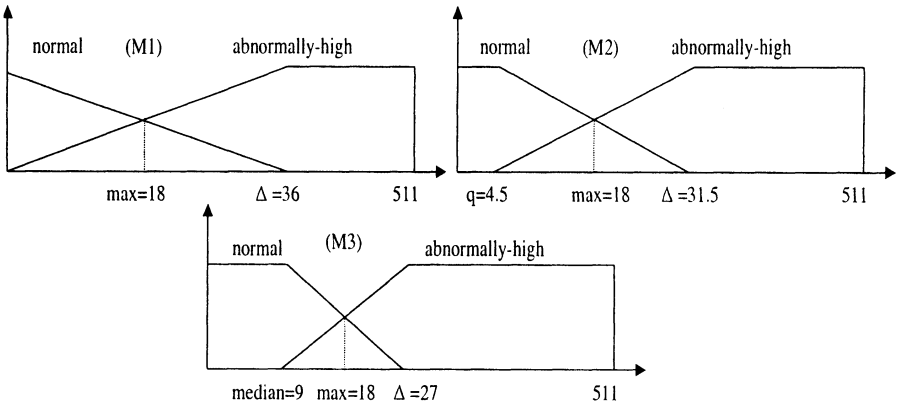

Figure 3. Membership functions for *same-host-diff-srv-in-time-window* variable. **M1**: minimum method **M2**: first quartile method **M3**: median method.

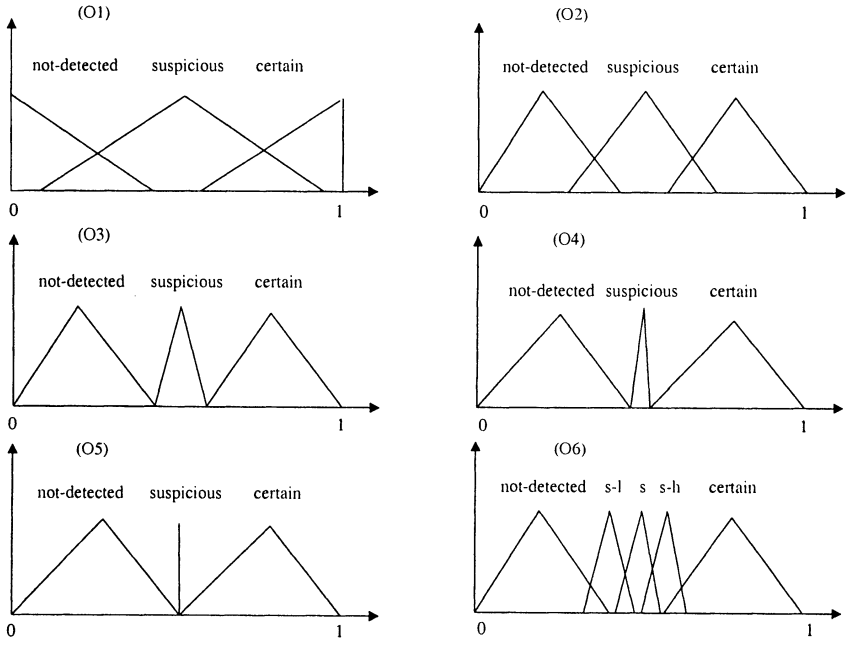

Figure 4. Output variable.

3. Simulation

We have carried out simulation studies using some of the features of the KDD CUP 1999 data set (KDD99). As an example, to obtain the

same-host-syn-error-in-time-window we have used all the records with flag *S0* during normal operation. These records show all instances of SYN packets. Feature 25 of the data set is used to obtain the percentage of packets with SYN error received by a host in a time window of 2 seconds. The number of connections to the same host with SYN error is obtained by multiplying each rate by the value of feature number 23. Note that feature 23 shows the total number of connections to the same host in the past 2 seconds. A monitor agent uses these statistics to build the normal membership function for *same-host-syn-error-in-time-window.*

3.1 Experimental Results

Table 2 shows the results of detecting general probing, smurf and neptune attacks. The results shown in Table 2 are based on using input and output membership functions similar to those shown in Figures 3(M1) and 4(O3), respectively.

The primary goal of monitor agent is to build the membership functions so that false positives can be avoided. False positives that are reported in Table 2 are due to the similarity among different types of attacks. For example, the 1900 no-neptune records reported by neptune detection agent as neptune are in fact probing records. There are no normal records among these 1900 records. No-smurf records (36 records) reported by smurf detection agent as smurf are Ping of Death (PoD) attack records. Due to the fragmentation error in these records, PoD detection agent reports these records as PoD attack with higher confidence. This will convince the decision agent that this in fact is a PoD attack and not a smurf one. Meanwhile, it is possible that a record belongs to two different attack groups with the same degree. Decision agent takes care of both and updates its belief based on the future reports from the detection agents.

Table 3 shows the results of our experiments using the output functions shown in Figure 4(O6) and different membership functions shown in Figure 3. It is seen that the relative population of probe records increases as we move from suspicious(low) to suspicious and suspicious(high) membership functions. The decision agent uses this information to detect the escalation of an attack. If the level of suspicion of a detection agent increases continuously, it causes a decision agent to believe that an attack is underway. On the other hand, steady suspicion even in the same level can be a sign of some unknown malicious activity in the system.

Table 2. The results of detecting general probing, neptune and smurf in the KDD CUP data.

Probe Detection Agent	*Label on the Data Set*	*Number of Matches*
not-detected	no-probing	4197474
not-detected	probing	6494
suspicious	no-probing	659845
suspicious	probing	13099
certain	no-probing	9
certain	probing	21509

Neptune Detection Agent	*Label on the Data Set*	*Number of Matches*
not-detected	no-neptune	473+4028601[a]
not-detected	neptune	880
suspicious	no-neptune	10
suspicious	neptune	440
certain	no-neptune	1900
certain	neptune	866126

Smurf Detection Agent	*Label on the Data Set*	*Number of Matches*
not-detected	no-smurf	3701+2086770[b]
not-detected	smurf	42
suspicious	no-smurf	37
suspicious	smurf	71
certain	no-smurf	36
certain	smurf	2807773

[a] records without *S0* flag.
[b] non ecr-i records.

Table 3. The results of probe detection with three levels of suspicions (Figure 4(O6)).

Probe Detection Agent	*Label*	*M1*	*M2*	*M3*
not-detected	no-probing	3684238	4082315	4259872
not-detected	probing	3988	5700	8002
suspicious(low)	no-probing	724019	443298	403913
suspicious(low)	probing	6102	4719	2774
suspicious	no-probing	449062	331706	193534
suspicious	probing	8905	8382	7830
suspicious(high)	no-probing	0	0	0
suspicious(high)	probing	1203	810	599
certain	no-probing	9	9	9
certain	probing	20904	21491	21897

3.2 Evaluation

Detection and false positive rates are the two major criteria for evaluating the performance of an intrusion detection system. To evaluate FAST, we have used a new criterion called false suspicious rate as well as a new definition for the detection rate. Let I, DI and DS represent the total number of intrusion instances in the test data, the number of intrusion instances correctly detected and the number of intrusion instances detected as suspicious, respectively. Let N represent the total number of normal instances in the test data, IN denote the number of normal instances incorrectly classified as intrusions, SN represent the number of normal instances incorrectly classified as suspicious and NI represent the number of intrusion instances that incorrectly classified as normal. System's performance is measured using the following ratios:

Conventional detection rate	$\mathcal{D}_1 = DI/I$
New detection rate	$\mathcal{D}_2 = (DI + DS)/I$
False positive rate	$\mathcal{F}_p = IN/N$
False suspicious rate	$\mathcal{F}_s = SN/N$
False negative rate	$\mathcal{F}_n = NI/I$

To evaluate the system, we examine the results obtained from the simulation of probe detection agent (see Table 4). Two different approaches are used to detect probes: time based and connection based. Time based window is used to detect fast probes. The second approach is based on a window of fixed number of recent connections (slow probe). We converted the relevant features in the KDD CUP data from the percentage format to the integer values and used them during training and test.

Table 4. The number of probing and no-probing in the KDD CUP data set.

Record Type	*Number*
probing records (*nnmap*, *portsweep*, *ipsweep*, *satan*)	41102
no-probing records (including all of the other intrusion records)	4857328
total	4898430

Figure 5 shows the conventional detection rate D_1 and the new detection rate D_2 of the system with different membership and output functions. It demonstrates the trade-off between D_1 and D_2. Membership functions with wide boundaries make the system more sensitive to suspicious activity and increase the new detection rate, D_2. At the same time, confidence of system for labelling the suspicious activities as attack decreases.

Figure 6 illustrates the false suspicious and false negative rates. It is seen that there is a trade-off between false negative and false sus-

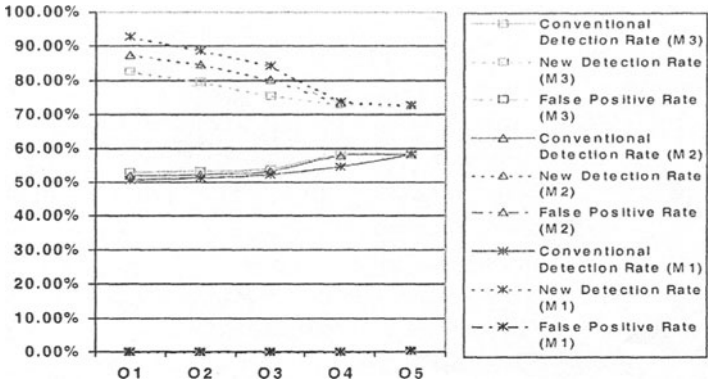

Figure 5. Trade-of between the conventional detection rate, $\mathcal{D}_1$, and the *new* detection rate, $\mathcal{D}_2$.

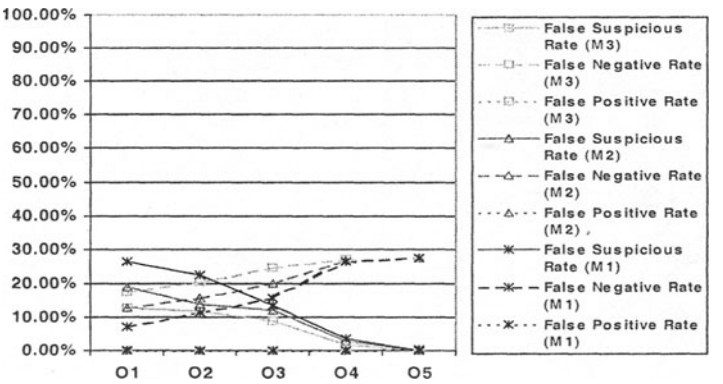

Figure 6. Trade-of between the false negative rate, $\mathcal{F}_n$, and the false suspicious rate, $\mathcal{F}_s$.

picious rates. In other word, to avoid false negatives, agents have to be suspicious to more records. The number of false positives increases slightly when we narrow down the suspicion interval. This is due to the fact that some attacks have similar effects and cause the rules from different detection agents to fire. These false positives don't label any benign connection as an attack, instead, they mistakenly label one type of attack with another one. When the suspicious interval is wide, most of these alarms are categorized as false suspicious. As we narrow down the suspicious interval, these records move from suspicious interval to the attack interval and increase the false positive rate. Decision agent solves this problem by assigning the record to the most possible attack category.

4. Conclusions

The main drawbacks of the current intrusion detection systems are: 1) large number of false positives, 2) inability to detect unknown attacks and 3) inability to properly assess the relative danger of the misuse and provide appropriate response. A new intrusion detection system for avoiding false positives and managing false negatives is proposed in this paper. The key elements of the proposed system are intelligent agents and fuzzy reasoning. Intelligent agents are capable of analyzing a situation, making decisions and communicating with other agents and users whereas fuzzy reasoning is used to identify and prioritize different attacks.

A prototype of the proposed multiagent-based intrusion detection system is simulated. The results obtained from the simulation studies using KDD CUP data set show that, through the selection of suitable membership functions for network variables, false positives can be avoided. Similarly, suitable output functions can significantly reduce false negatives at the cost of increasing the false suspicions. The system's performance on the KDD CUP data set as illustrated in Section 3, indicates that the proposed system can successfully manage false alarms.

Acknowledgments

Mr. Shajari's work is supported by the National Research Council, Fredericton, NB, Canada. Dr. Ghorbani's work is partially supported through grant RGPIN 227441-00 from Natural Science and Engineering Research Council of Canada.

References

Dubois, D. and Prade, H. (1993). Fuzzy sets and probability: Misunderstandings, bridges and gaps. *Proc. of the Second IEEE Inter. Conf. on Fuzzy Systems*, 2:1059-1068.

Dumitrescu D. (1993) Fuzzy Measures and Entropy of Fuzzy Partitions. *Journal of Mathematical Analysis and Applications*, 176:359-373.

Internet Security Systems Co. (2001). The Truth about False Positives. Available at http://documents.iss.net/whitepapers/TheTruthAboutFalsePositives.pdf

Jaulent M., Bombardier V., Cherrak I. and Perez-Oramas O. (2000). Fuzzy Quantification of Artery Lesions in Renal Arteriographies. In Szczepaniak, P. S., Editor, *Fuzzy Systems in Medicine*, pages 470-428. Physica-Verlag Company.

KDD CUP 1999 dataset Available at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Ross T. J. (2000). Membership Functions, Fuzzification and Defuzzification. In Szczepaniak, P. S., Editor, *Fuzzy Systems in Medicine*, pages 48-77. Physica-Verlag Company.