Perceptual Intrusion Detection System

Atsushi Inoue *
Department of Computer Science
Eastern Washington University
Cheney, WA 99004-2412 U.S.A.
E-mail: atsushi.inoue@ewu.edu

Abstract

A simple intrusion detection system (IDS) with respect to perception of human experts is proposed. Its computational framework is designed based on concepts of computational theory of perceptions (CTP) and mass assignment theory (MAT). CTP provides a computational framework of representing and handling perception using linguistic terms and corresponding fuzzy sets. MAT provides that of representing and handling a bias consisting in between perception and observed intrusions through the consistency management between fuzzy sets and probability distributions. For the knowledge construction of this IDS, only linguistic descriptions extracted from organization policies and perception of human experts is expected. For the inference and refinement of knowledge, Support Logic, a truth functional logic for manipulating probability intervals, is used.

1 Introduction

Intrusion detection has been recognized as a new trend in applied research of computer science. Tasks are performed based on analysis of patterns that intrusion possesses. More specifically, intrusion detections are most likely performed by one, or a combination, of the following two types of pattern recognition[1]:

1. *Anomaly recognition*: Recognition by the negation of patterns of normal activities (i.e. anomaly).
2. *Signature recognition*: Recognition of patterns possessed by a certain intrusion of your interest.

There are some fundamental trade-offs between these two types of intrusion detection. Anomaly detection is easier for obtaining samples, i.e. normal network activities. However, this does not have a capability of identifying specific intrusions. This imposes further investigations on computer networks as some noticeable anomaly is detected. On the other hand, signature detection specifically identifies the footprint of intrusions. The drawback is that it is not capable of noticing unknown (i.e. new) intrusions.

Recently, demands on security tools such as intrusion detection systems (IDSs) are significantly increased due to the exponential increase of malignant activities and shortage of manpower for network administrations. It is mandatory that network administrators respond quickly at all costs regardless of how shorthanded they are every time when some damages are made on the computer networks.

Conventionally, knowledge-based or rule-based approaches are dominantly used for intrusion detection tasks. However, knowledge constructions for intrusion detection, especially for signature recognitions, are not easy because of its high dimensionality and dependencies among them from analytical aspects (i.e. the cause of base-fallacy problems [2]) and because of its massive variety and the appearance of newly innovated intrusions with unusual rapidity. In practice, collection of intrusion samples is more difficult than that of normal activities in most of organizations.

We cannot expect a collection of ample samples of intrusions for the knowledge construction. However, we can still count on perception of experts, i.e. network administrators. In fact, successful intrusion detection tools such as *snort* [7] simply use intrusion patterns directly specified based on perceptions of experts (in regular expressions for snort). Alternatively, many competent network administrators utilize a simple network packet dumping tools such as *tcpdump* [8] in conjunction with or independent from conventional IDSs.

In this paper, we propose a simple knowledge-based IDS with respect to such perception of experts based on computational theory of perception (CPT) [9] and mass assignment theory (MAT) [3] (PIDS for short). The PIDS is designed

1. to be a *useful* utility such as snort and tcpdump,

*The Director of Inland Northwest Security Systems Initiative (INSSI) within Department of Computer Science at Eastern Washington University. This research is in part supported by the Congressional Appropriation for EWU's Technology Initiative for New Economy (TINE) and a NSF curriculum development grant (NSF 0230590).

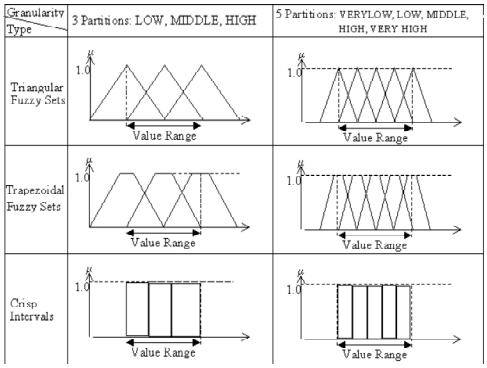

Figure 1. Fuzzy Partitions

2. with a *simple* system architecture,
3. to take *linguistic* descriptions from experts by following CTP,
4. with a *powerful* representation and inference by following the concept of Perceptual Information Processing (PIP) [4] [5] (the work of my Ph.D. dissertation), and
5. to be *computationally efficient* by using *Support Logic* and its implementation within a logic programming environment called *Fuzzy Relational Inference Language* (FRIL) [3].

This IDS is intended to be a type of intrusion detection agents incorporated within a framework of multi-agent intrusion systems called *SPIDER* [6].

2 Computational Model

By following CTP, perception is represented in terms of a linguistic representation and fuzzy information granulation. The inference is performed within the framework of fuzzy logic underlying on MAT.

2.1 Linguistic Description and Computational Perception

According to CTP, human experts usually represents a domain of values by some linguistic terms such as 'low', 'medium', and 'high'. A granulation, the fuzzy information granulation incorporated within CTP in particular, is determined by a set of such linguistic terms reflecting human's perception. Some examples of granulation for a set of integer $\mathcal{Z}$ can be given as follows (from a large granulation to a small one):

- any value (granularity of 1 – the entire domain)
- small, medium, large (granularity of 3).
- very small, small, medium, large, very large (granularity of 5).
- tiny, very small, small, medium, large, very large, extra large (granularity of 7).

  $\vdots$
- $1, 2, \ldots$ (granularity of $|\mathcal{Z}|$)

Please notice that the cardinality of such granulation approaches to that of the original domain as single granule in this granulation becomes smaller.

For a vector space (i.e. a Cartesian product of domains of values) $X_1 \times \ldots \times X_n$, the cardinality of granulation of the entire vector space is determined by $\prod_i |F_i|$ where a granulation F_i (i.e. a set of linguistic terms) for X_i is obtained for $1 \leq i \leq n$.

Let a set of classes be C. Assume that each and every instance x within this vector space belongs to a class $c \in C$ (i.e. x is c). Then a linguistic description of the class c as a collection of IF-THEN rules with granulations of all domains is given such that

$$\begin{array}{c} \text{IF } x_1 \text{ is } l_{11} \text{ AND} \ldots \text{AND } x_n \text{ is } l_{n1} \text{ THEN } x \text{ is } c \\ \vdots \\ \text{IF } x_1 \text{ is } l_{1i} \text{ AND} \ldots \text{AND } x_n \text{ is } l_{ni} \text{ THEN } x \text{ is } c \end{array}$$

where $x = (x_1, \ldots, x_n)$ is an instance (i.e. a vector) and l_{ij} is a linguistic term in granulation F_i where $1 \leq j \leq |F_i|$.

By providing granulations F_i over $X_i \; \forall \; 1 \leq i \leq n$, class c can be represented (i.e. knowledge of c can be acquired) from human experts in linguistic descriptions.

Now we consider assigning fuzzy partitions, collections of fuzzy sets defined over a domain which correspond to linguistic terms describing granular shown in Figure 1, to all granulations F_i for the purpose of computations (i.e. inferences and tuning). Formally, let

$$\begin{array}{rll} P_i = \{< l_{ij}, \mu_{ij} > \mid & l_{ij} \in F_i \\ & \wedge \text{MAX}_{e \in X_i}[\mu_{ij}(e)] = 1 \\ & \wedge \sum_j \mu_{ij}(e) = 1 \; \forall \; e \in X_i\} \end{array}$$

be a fuzzy partition assigned to a granulation F_i (note: P_i consists of normalized fuzzy sets). Then hypotheses of each IF-THEN rule represents a single granule within the vector space, and the entire rule represents a cluster of instances of class c. Notice that this granule is a fuzzy granule, i.e. a Cartesian product of fuzzy sets defined over dimensions of the input vector space. Notice also that such a IF-THEN rule does not allow a blend of instances belonging to different classes at all. In other words, it is assumed that

$$Prob(x \text{ is } c \mid \bigwedge_i x_i \text{ is } l_{ij}) = 1$$

To allow such a blend, the collection of IF-THEN rules should be extended with a probability

$$p_k = Prob(x \text{ is } c_k \mid \bigwedge_i x_i \text{ is } l_{ij})$$

such that $\sum_k p_k = 1$ where p_k is a normalized frequency of instances belonging to class c_k within a granule represented by single IF-THEN rule. Consequently, such a collection is given such that

$$\begin{array}{c} \text{IF } x_1 \text{ is } l_{11} \text{ AND } \ldots \text{ AND } x_n \text{ is } l_{n1} \text{ THEN } x \text{ is } c \text{ is } p_1 \\ \vdots \\ \text{IF } x_1 \text{ is } l_{1i} \text{ AND } \ldots \text{ AND } x_n \text{ is } l_{ni} \text{ THEN } x \text{ is } c \text{ is } p_i \end{array}$$

Such a probability may not be obtained as a crisp number because the count of instances belonging to a fuzzy granule may not necessarily be crisp. As a matter of fact, it is crisp if all instances within a granule belong to the subspace represented as Cartesian product $f^1_{1i} \times \ldots \times f^1_{ni}$ where f^1 is a crisp set such that $\mu_f(x) = 1 \; \forall \; x \in f^1$ (a.k.a. α-cut of a fuzzy set f with $\alpha = 1$). Formally, the consequence of the above rule may have a fuzzy probability, a fuzzy set defined over $[0, 1]$ (e.g., 'x is 'DoS' is '*highly probable*'). In summary, the probability p within a consequence of a IF-THEN rule can be either one of the following:

1. Point probability: $p \in [0, 1]$
2. Support pair (i.e. an interval of probability): a pair of lower bound p_l and upper bound p_u of probability such that $p = (p_l, p_u) \in [0, 1] \times [0, 1]$
3. Fuzzy probability: $p = f_p$ where f_p is a fuzzy set characterized by its membership function $\mu_{f_p} : [0, 1] \mapsto [0, 1]$.

We are currently investigating whether there is the following correlation between the probability p within a consequence 'x is c is p' and the location of the blend of instances within a granule (or not):

1. Point probability: All instances needs to be located at $f^1_{1i} \times \ldots \times f^1_{ni}$.
2. Support pair: All instances needs to be located at $f^\alpha_{1i} \times \ldots \times f^\alpha_{ni}$ where $0 < \alpha < 1$.
3. Fuzzy probability: instances are located within $f_{1i} \times \ldots \times f_{ni}$.

This may be justified by the representation (decomposition) theorem ($f = \bigcup_{\alpha \in [0,1]} \alpha \cdot f^\alpha$) and MAT (ongoing work).

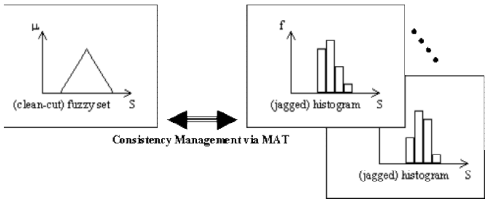

Figure 2. Consistency Management via MAT

2.2 Consistent Uncertainty Management

MAT is introduced in order to handle more semantics of data in the sense of unification operation in logic programming. Feasibility of MAT as a basis for representing and handling perception has been studied (i.e. PIP) and a short text classification application has been successfully developed.

Let S be a sample space. Then a *mass assignment* (MA) m_S associated with S is a function from the power set $\mathcal{P}(S)$ to an interval of real numbers such that $m_S : \mathcal{P}(S) \mapsto [0, 1]$ and $\sum_{A \subseteq S} m_S(A) = 1$. A subset $A \subseteq S$ is called a *focal element* for mass assignment m_S if $m_S(A) > 0$.

MAT provides the following correspondences among probability distributions, mass assignment, and fuzzy sets:

1. *MA and probability*:

$$P_S(x) = \sum_{A \subseteq S, x \in A} P_A(x) \cdot m_S(A) \qquad (1)$$

where P_S is a probability distribution on S, m_S is a mass assignment over S and P_A is a probability distribution on A (often called a selection rule). The selection rule represents bias (i.e. preference) on elements within A. The selection rule without any bias is $P_A(x) = \frac{1}{|A|}$ (the least prejudged distribution).

2. *MA and fuzzy sets*: Let $F = x_1/\mu_1 + \cdots + x_n/\mu_n$ be a fuzzy subset over S. We denote $\mu_i = \mu_F(x_i)$ and without loss of generality we assume

$$1 = \mu_1 \geq \cdots \geq \mu_n \geq \mu_{n+1} = 0$$

Then a MA with nested focal elements $\{x_1, \ldots, x_i\}$ for $i = 1, \ldots, n$ can be derived as

$$m_S(A) = \begin{cases} \mu_i - \mu_{i+1} & \text{if } A = \{x_1, \ldots, x_i\} \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

Note that this can be rewritten by following the representation theorem such that

$$m_S(A) = \begin{cases} \alpha_i - \alpha_{i+1} & \text{if } F^{\alpha_i} = \{x_1, \ldots, x_i\} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Table 1. Example of Meet Operation: $K(A,B) = m(A) \cdot m(B)$

$m_1 \wedge m_2$	$\{b\}:0.3$	$\{a,c\}:0.4$	$\{a,b,c\}:0.3$
$\{a\}:0.2$	$\phi:0.06$	{a}:0.08	{a}:0.06
$\{a,b\}:0.5$	$\{b\}:0.15$	$\{a\}:0.2$	$\{a,b\}:0.15$
$\{b,c\}:0.3$	$\{b\}:0.09$	$\{c\}:0.12$	$\{b,c\}:0.09$

Table 2. Example of Join Operation: $K(A,B) = m(A) \cdot m(B)$

$m_1 \vee m_2$	$\{b\}:0.3$	$\{a,c\}:0.4$	$\{a,b,c\}:0.3$
$\{a\}:0.2$	$\{a,b\}:0.06$	$\{a,c\}:0.08$	$\{a,b,c\}:0.06$
$\{a,b\}:0.5$	$\{a,b\}:0.15$	$\{a,b,c\}:0.2$	$\{a,b,c\}:0.15$
$\{b,c\}:0.3$	$\{b,c\}:0.09$	$\{a,b,c\}:0.12$	$\{a,b,c\}:0.09$

3. *Probability and fuzzy sets*: The mapping between fuzzy sets and probability distributions via a MA is obtained from above two such that

$$P_S(x_k) = \sum_{i=k}^{n} P_A(x_k) \cdot (\mu_i - \mu_{i+1}) \qquad (4)$$

There are operations within MAT defined in a way compatible to set operations such as *complement* ($\tilde{}$), *meet* ($\wedge$), and *join* ($\vee$). Here, we define these operations in terms of a function $K(A,B)$ such that $K : \mathcal{P}(S) \times \mathcal{P}(S) \mapsto [0,1]$ satisfying the following constraints inherited from properties of mass assignment formerly defined:

1. $\sum_A K(A,B) = m(B)$
2. $\sum_B K(A,B) = m(A)$
3. $\sum_{A,B} K(A,B) = 1$
4. $K(A,B) = 0 \; \forall A \forall B \; m(A) = 0 \vee m(B) = 0$

Then (general) definitions of these operations are given as follows:

1. *Meet*: $m_1 \wedge m_2 : \mathcal{P}(S) \mapsto [0,1]$ such that

$$m_1 \wedge m_2(C) = \sum_{C=A \cap B} K(A,B) \qquad (5)$$

2. *Join*: $m_1 \vee m_2 : \mathcal{P}(S) \mapsto [0,1]$ such that

$$m_1 \vee m_2(C) = \sum_{C=A \cup B} K(A,B) \qquad (6)$$

3. *Complement*: $\overline{m} : \mathcal{P}(S) \mapsto [0,1]$ such that

$$\overline{m}(A) = K(A, S-A) = m(\overline{A} = S - A) \qquad (7)$$

As you see, K for the complement is determined uniquely (i.e. $K(A, S-A) = m(\overline{A})$) in order to be consistent with the original definition of MAT operations. Please notice that this may be extended by selecting an appropriate K to classes of fuzzy complement such as Sugeno class and Yager class. Please notice that results of meet and join operations vary in the selection of K. Tables 1 and 2 shows examples of meet and join are shown with $K(A,B) = m(A) \cdot m(B)$ (meet and join by multiplication). Please also notice that the operation of Combining evidences from Dempster-Shafer theory of evidence can be defined in terms of this meet operation such that

$$m_1 \otimes m_2(C) = \begin{cases} \frac{m_1 \wedge m_2(C)}{1 - m_1 \wedge m_2(\emptyset)} & \text{if } C \neq \emptyset \\ 0 & \text{if } C = \emptyset \end{cases}$$

In addition, it is interesting to investigate further on correlations between the selection of K and t-norm and t-conorm (future work).

2.3 Inference

For inference that fuzzy sets are involved (e.g., the one for PIDS), comparison of two fuzzy sets, say f and g, is necessary. Such operation within Support Logic is called *Semantic Unification* [3], denoted as $m_f | m_g$, which replaces the symbolic unification operation defined within logic programming (e.g., PROLOG). Formally, Semantic Unification can be defined as a function $(\mathcal{P}(S) \mapsto [0,1]) \times (\mathcal{P}(S) \mapsto [0,1]) \mapsto (\{t,u,f\} \mapsto [0,1])$, for given two mass assignments m_f and m_g corresponding to fuzzy sets f and g respectively, that determines a probability distribution over truth values, $P(t)$, $P(u)$ and $P(f)$, such that

$$m_f | m_g = \begin{cases} P(t): \sum_{B \subseteq A} K(A,B) \\ P(u): \sum_{A \cap B \neq \emptyset} K(A,B) \\ P(f): \sum_{A \cap B = \emptyset} K(A,B) \end{cases} \qquad (8)$$

Table 3. Example of Semantic Unification: $K(A,B) = m(A) \cdot m(B)$

$m_f \vert m_g$	$\{b\}:0.3$	$\{b,c\}:0.5$	$\{a,b,c\}:0.1$	$\{a,b,c,d\}:0.1$
$\{a\}:0.3$	$f:0.09$	$f:0.15$	$u:0.03$	$u:0.03$
$\{a,b\}:0.5$	$t:0.15$	$u:0.25$	$u:0.05$	$u:0.05$
$\{a,b,c\}:0.2$	$t:0.06$	$t:0.1$	$t:0.02$	$u:0.02$

Table 3 shows an example. We then obtain a support pair of the unification from the above probability distribution P such that

$$(p_l, p_u) = (P(t), P(t) + p(u)) = (P(t), 1 - P(f)) \quad (9)$$

It is shown that there is a consistency between Support Logic and Fuzzy Logic, particularly these for fuzzy control (i.e. crisp input values).

For a single IF-THEN rule and an input vector $x = (x_1 = v_1, \ldots, x_n = v_n)$, the inference is represented as

IF x_1 is f_1 AND ... AND x_n is f_n THEN x is c is f_p
x_1 is v_1
$\vdots$
x_n is v_n
---
x is c is p

where $f_i,\ 1 \leq i \leq n$, are fuzzy sets associated with linguistic terms and $v_i,\ 1 \leq i \leq n$, are input values (crisp values or fuzzy values). The truth value of the consequence 'x is c is p' is determined as a support pair in Support Logic such that

$$(p_l, p_u) = (\prod_{i=1}^{n} p_{l_i}, \prod_{i=1}^{n} p_{u_i}) \quad (10)$$

where the i-th support pair is obtained by semantic unification $m_{f_i} \vert m_{v_i}$, and m_{f_i} and m_{v_i} are mass assignments corresponding to fuzzy sets f_i and v_i (a singleton if it is a crisp input) respectively. By following the concept of *expected fuzzy set* [3], the membership function of the fuzzy probability p is computed from f_p and the support pair of the truth value (p_l, p_u) such that

$$\mu_p(y) = p_l \cdot \mu_{f_p}(y) + (1 - p_u) \cdot (1 - \mu_{f_p}(y)) + (p_u - p_l) \quad (11)$$

In case of multiple rules deriving consequences for the same class, results of these inferences need to be disjunctively combined such that $\bigcup_j p_j$. Alternatively, such a combination can be viewed as a matter of aggregating decisions made by multiple decision-making agents such as the decision making model within SPIDER underlying MAT [6]. As a matter of fact, SPIDER decision making model is a general framework subsuming the case of fuzzy set disjunction.

For the defuzzification of the (combined) result, there are following options:

1. *Use of SPIDER decision making model*: Both defuzzification and the combination of inference results may be handled in an integrated manner as the SPIDER decision making model takes various types of decisions: (point) probabilities, support pairs and fuzzy probabilities.

2. *Centroid method* dominantly used for fuzzy control.

3. *α-cut of p*: $\text{ARGMAX}_{\mu_p(y)=\alpha\ \forall y \in [0,1]}[p^\alpha]$ gives the necessitated support pair. Optionally, the mid point can be taken as the defuzzification for a point probability.

2.4 Refinement of Perception

As PIDS is in use, collections of inputs and their corresponding results can be obtained and summarized as frequency histograms. As mentioned earlier, there is a consistency between probability distributions (i.e. normalized frequency histograms) and fuzzy sets within partitions of granulations corresponding to linguistic terms used in IF-THEN rules (see Figure 2). Then, as a result of studying on PIP [4], the following consistency management schemes in terms of MA, a corresponding fuzzy set (FS), a corresponding probability distribution (PD) and a prior probability representing a bias (selection rules; SR for short) have been developed:

1. *Point Mapping* Determines a correspondence between FS and PD with a fixed SR (bias).

2. *Band Mapping* Determines a family of FS (PD) for a fixed PD (FS) by tuning parameters of SR in various ways.

3. *Bidirectional Associative Memory Model* [5].

Here a simplified model of PIP is proposed by parameterizing fuzzy sets within a fuzzy partition such that $\mu(x) = f_{[a,b,c,d]}(x)$ satisfying the following

1. $a \leq b \leq c \leq d$ (point, interval, triangular or trapezoidal)

2. $c_i = a_{i+1}$ and $d_i = b_{i+1}\ \forall\ 1 \leq i \leq n$ partition $- i$ (constraints of fuzzy partition)

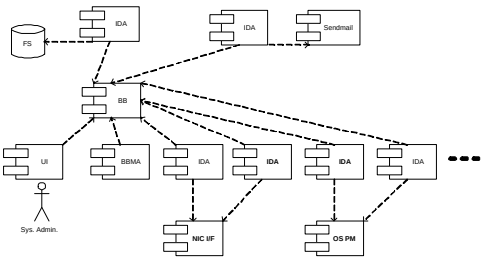

Figure 3. Example of SPIDER Architecture

3. $\mu(a) = \mu(d) = 0$ (i.e. *support*) and $\mu(x) = 1 \forall x \in [b, c]$ (i.e. *core*) (normalized)

4. $\mu(x) = \frac{x-a}{b-a} \forall x \in (a, b)$, $\mu(x) = \frac{d-x}{d-c} \forall x \in (c, d)$ (linearly interpolated)

With this parameterization, a granulation (i.e. a fuzzy partition) consists of a sequence of fuzzy (i.e. $\mu(x) < 1$) and core ($\mu(x) = 1$) partitions. Considering the corresponding probability distributions, any probabilities (i.e. normalized frequencies) within the core partitions need to be higher and crispier than those within the fuzzy partitions. More formally, Consider a MA m and a constant ϕ, representing the number of α-cuts within the corresponding fuzzy set, such that

$$m(A_i = [b-(i-1)\cdot\frac{b-a}{\phi}, c+(i-1)\cdot\frac{d-c}{\phi}]) = \frac{1}{\phi} \quad (12)$$

$\forall\ 1 \leq i \leq \phi$. Then parameters b and c are adjusted subject to

$$\forall x \in A_i \forall y \in A_{i+1} - A_i, P(x) \geq P(y) \quad (13)$$

This approach guarantees simple shapes of fuzzy sets consistent with jagged histograms (i.e. the elasticity of fuzzy sets) and is advantageous in aspects of computation and overfitting (further analysis follows).

3 System Architecture and Implementation

SPIDER architecture consisting of multiple autonomous agents such as intrusion detection agents (IDAs), a blackboard agent (BB) and a blackboard manager agent (BBM) is currently used (Figure 3). For sensors to generate inputs, we currently study on tcpdump to extract network packets (TCP) and word frequencies of E-mail body (EMAIL). Here are the current configurations of PIDS:

1. Single-class, single-rule IDAs with the tcpdump sensor for real use and KDDCup99 data set (40+ features) for testing.

2. Single-class, two-rule (one representing class signatures and the other representing anomaly) IDAs with the tcpdump sensor for real use and KDDCup99 data set (40+ features) for testing.

3. BB and BBM for aggregating decisions.

4. FRIL (for inference) and C++ (for integration).

Currently, we obtain the best results of over 95% of accuracy and below 10% of fallout (i.e. false alarms).

4 Summary and Future Works

A simple knowledge-based intrusion detection system, PIDS, is introduced. In addition to analytical studies mentioned formerly, the optimized identification of input vector space and a hardware implementation is currently being planned. In particular, the single input rule module (SIRM) architecture is being investigated as the simplest case.

References

[1] S. Axelsson, *Intrusion detection systems: A taxonomy and survey*, Technical Report 99-15, Dept. of Comp. Engr., Chalmers University of Technology, Sweden, 1999.

[2] S. Axelsson, *On a difficulty of intrusion detection*, Proc. Workshop on Recent Advances in Intrusion Detection, 1999.

[3] J. F. Baldwin, T. P. Martin, B. W. Pilsworth, *FRIL-Fuzzy and Evidential Reasoning in Artificial Intelligence*, Research Studies Press,1995.

[4] A. Inoue, A. L. Ralescu, *Computational Model of Perceptual Information Processing*, Proc. FUZZ/IEEE99, pp. 824-829, Seoul, South Korea, 1999.

[5] A. Inoue, A. L. Ralescu, *The Associative Nature of Perceptual Information Processing*, Proc. ANNIE99, pp. 785-790, St. Louis, MO, 1999.

[6] P. Miller, J. Mill, A. Inoue, *Synergistic and Perceptual Intrusion Detection with Reinforcement (SPIDER)*, MAICS03, Cincinnati, OH, 2003.

[7] `http://www.snort.org/`, 2003.

[8] `http://www.tcpdump.org/`, 2003.

[9] L. A. Zadeh, *From Computing with Numbers to Computing with Words: From Manipulation of Measurements to Manipulation of Perceptions*, IEEE Tranc. Circuits and Systems I, vol. 45, No. 1, pp. 105-119, 1999.