Computer Assisted Mechanics and Engineering Sciences, 14: 579–590, 2007.

Dual representation of samples for negative selection issues

Andrzej Chmielewski
Białystok Technical University, Faculty of Computer Science
Wiejska 45A, 15-331 Białystok, Poland

Sławomir T. Wierzchoń
Institute of Computer Science, Polish Academy of Sciences, and
Gdańsk University, Faculty of Mathematics, Physics and Informatics, Poland

(Received in the final form September 18, 2007)

This paper presents a new dual model combining binary and real-valued representations of samples for negative selection algorithms. Recent research show that the two types of encoding can produce quite good results for some types of datasets when they are applied separately in such algorithms. Besides a number of efficient algorithms, various affinity (or similarity) functions fitted to particular implementation was investigated. Basing on a series of experiments, we propose a dual representation enabling overcome some of the existing drawbacks of these algorithms, and allowing significant speed up the classification process. This new model was designed mainly for detecting anomalies in real-time applications, were the time of classification is crucial, e.g. intrusion detection systems.

Keywords: anomaly detection, negative selection, binary receptors, real-valued receptors, intrusion detection

1. Introduction

One of the crucial problems in developing automated classification systems is the choice of appropriate representation for samples gathered in the datasets. Typically, most of the samples are encoded as vectors of real-valued numbers. Unfortunately, this type of representation can involve the use of many time consuming operations (like multiplication and division), e.g. when classification process have to compute a distance between two samples to measure the similarity between them. Therefore, in the case of huge dataset containing tens of attributes, the choice of this type of representation seems to be far from optimal, especially when the time of classification is crucial.

As an example imagine a network-based Intrusion Detection Systems (IDS) which must be capable to perform real-time traffic analysis to detect unacceptable system and network activity.

A classical IDSs uses the set of signatures (described even by more than 30 attributes) that define what suspicious traffic is. Thus, to detect intruders, every network connection is compared (attribute by attribute) against signatures gathered in the database. For example, Snort [21], the most popular *free software* IDS, uses the following rules:

```
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"DDOS Stacheldraht
gag server response"; icmp_id:669; itype:0; content:"sicken";
reference:arachnids,195; classtype:attempted-dos; sid:225; rev:6;)
```

Since the time complexity of comparing attribute's values, regardless of used representation (string, integer or real value), is very low (as opposed to multiplications and divisions), the type of representation is not crucial. However, this approach losses its effectiveness when the signatures (or rules) are written to identify precisely each currently known issue. As a result, the database with

signatures grows and grows as well as new (unseen) and variations of known attacks are not fully detected.

A possible solution to this problem relies upon generalization of too stringent rules. Through the relaxing and varying the conditions and parameters, it is possible to decrease the number of rules (faster classification) as well as to identify novel attacks. On the other hand, too relaxed constraints can lead to raising the alarms for legitimate network connections. This problem, for Snort's rules, is widely discussed in [3].

An interesting alternative for classical approaches to intrusion detection problem is offered by Artificial Immune Systems (AIS). This approach is inspired by the Natural Immune System (NIS) recognized as one of the most complex system in Nature. It is responsible for protecting an organism against damage from extremely large number of harmful bacteria, viruses, parasites and fungi, termed *pathogens.* To detect and eliminate pathogens, which constantly attack the organism, the NIS continually generates different receptors which tolerate own cells and react only when they meet any pathogen. The process of tuning receptors towards the attackers is called *affinity maturation.* This is the very important process as in its result we obtain a set of receptors which do not recognize any own cell, and almost perfectly react against intruders. It is worth to notice, that only own cells are used during the maturation and, what is characteristic for the NIS, no negative examples (descriptions of pathogens) are necessary. Therefore, mature receptors are able to recognize even previously not met foreign bodies, which total number is far greater than 10^{16} whereas the total number of own cells has been estimated to be about 10^6. This great efficiency, achieved by huge diversity of receptors, is the important reason of interest in developing AIS observed in recent years.

A reader interested in detailed description of the immune mechanisms is referred to e.g. [20, 30].

Since there, many immune-inspired IDS were presented (see [2] for review). They are implemented through the use of a self-nonself model, where *self* denotes own cells and *nonself* denotes pathogens. Most of these systems use negative selection principle (see Section 2) to generate receptors which can be compared to generalized rules of classical IDSs. Each receptor is capable to recognize many types of nonself samples (e.g. not legitimate network connections) by calculating its similarity to a given (i.e. censored) sample. Depending on used representation (binary or real-valued) different affinity functions (typically measuring a distance between compared objects) can be regarded. The advantages and disadvantages of both representation with examples of different algorithms of generating detectors, we discuss in Section 4. In Section 4.3, we present new dual model of representing the samples which combine both the encodings to achieve higher efficiency and to speed up the classification time.

2. Negative selection

One of the major algorithms developed within emerging field of AIS is Negative Selection Algorithm, proposed by Forrest *et al.* [12]. It is inspired by the process of thymocytes (i.e. young T-lymphocytes) maturation. To be more formal, denote U the problem space, or Universe of discourse, (e.g. U is a set of all possible strings of fixed length, see Section 4.1), and let $S \subset U$ be a subset of strings representing typical behavior of a system under considerations. Then the set of elements characterizing anomalous behavior, N can be viewed as the set-theoretical complement of S,

$$N = U \backslash S. \tag{1}$$

The elements of S are called *self*, and those of N are termed as *nonself.*

The negative selection algorithm relies upon generation of so-called detectors in such a way, that a freshly generated detector d is added to the set D of valid detectors only if it recognizes at least one element in N, and does not recognize any self element. In simplest case the detectors are generated randomly, but smart techniques are requested in general [12]. To mimic the process of self/nonself recognition we must designate a rule, $match(d, u)$, specifying when a detector d activates given an element u, see e.g. [29] for details. Usually, $match(d, u)$ is modeled by a distance metric or

a similarity measure [13]. Majority of detection rules induce so-called *holes*, i.e. regions of N which are not covered by any detector and therefore samples from this region will be classified as self.

In real-life applications, usually, it is not possible to gather all elements of S, because typically only small its subset is observed. Therefore, it is assumed, that S is composed of S_{seen} and S_{unseen},

$$S = S_{seen} \cup S_{unseen}, \tag{2}$$

but only S_{seen} is taken into the training phase (i.e. the phase of detectors generation). As a result, the detectors from the set D can recognize not only nonselfs from $N' \subset N$, but also some elements from S_{unseen}. The existence of *holes* in S_{unseen} is necessary to generalize beyond the training set (see [24] for details).

3. Testing dataset

To design efficient immune-based IDS we need to develop algorithms which are able to recognize the intruders described by tens of attributes. Therefore we choose the KDD Cup 1999 [14] as a test set; it is very popular dataset used to evaluate ID algorithms as well as other classification algorithms. It contains almost 5 millions records. Each record describes a network connection as a vector of 41 attributes (38 numerical and 3 symbolic) with additional end-label (`normal` value means self connection).

Examples of self and nonself connections are given below:

91, udp, domain_u, SF, 87, 45, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 147, 140, 0.95, 0.01, 0.18, 0.00, 0.00, 0.00, 0.00, 0.00, **normal**.

0, icmp, eco_i, SF, 18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 1, 153, 1.00, 0.00, 1.00, 1.00, 0.00, 0.00, 0.00, 0.00, **ipsweep**.

In our experiments all redundant connections were eliminated first. As a result we obtained reduced set S_{unique} containing 1 074 994 (almost fivefold reduction) unique connections (812 813 normal and 262 181 anomalous).

4. Samples representations

As already mentioned, to model the interactions (matching) between self and/or nonself elements we need three elements: (a) proper samples encoding, (b) appropriate affinity function, and (c) appropriate algorithm enabling generation of the receptors. In this paper, we consider two basic encodings (representations) recently used for negative selection algorithm purposes: binary and real-valued. For both of them many algorithms generating detectors were designed to achieve possibly highest detection rates with various similarity measurements functions. Short review of current state in this domain is presented below (see also [2]).

4.1. Binary representation

In case of binary encoding, we identify the Universe with l-dimensional Hamming space, $\mathbb{H}^l$ build out of all binary strings of fixed length l,

$$\mathbb{H}^l = \{\underbrace{000\ldots000}_{l}, \underbrace{000\ldots001}_{l}, \ldots, \underbrace{111\ldots111}_{l}\}.$$

4.1.1. Binary matching functions

Percus *et al.* [19] proposed the r-contiguous matching rule for abstracting the similarity between two strings in $\mathbb{H}^l$. According to this rule, two samples $h_i, h_j \in \mathbb{H}^l$, match if at least r contiguous bits in both strings are identical. Below an example of matching a sample by a detector for affinity threshold $r = 3$ is given

$$\overbrace{1\,0\,0\,0\,1\,1\,1\,0}^{l} \text{ sample,}$$
$$0\,1\,\underbrace{0\,0\,1}_{r}\,0\,0\,1 \text{ detector.}$$

The number of possible elements recognized by a detector $d \in \mathbb{H}^l$ with given threshold value r is $(l - r + 1) \cdot 2^{l-r}$, i.e. $(8 - 3 + 1) \cdot 2^{8-3} = 6 \cdot 32$ in our case. With a given detector d can associate the set of $l - r + 1$ templates; for instance, if $d = \{1\,1\,0\,0\,1\,0\,0\,1\}$, it recognizes all samples of the form $\{1\,1\,0\,*\,*\,*\,*\,*\}$, $\{*\,1\,0\,0\,*\,*\,*\,*\}$, $\{*\,*\,0\,0\,1\,*\,*\,*\}$, $\{*\,*\,*\,0\,1\,0\,*\,*\}$, $\{*\,*\,*\,*\,1\,0\,0\,*\}$, $\{*\,*\,*\,*\,*\,0\,0\,1\}$, where the asterisk * is the wild-card symbol representing either 0 or 1. The threshold r is chosen in advance. When choosing this value, one should remember that the greater r is, the most specific the detectors become.

A variant of the r-contiguous bits rule is the r-chunk matching rule proposed by Balthrop *et al.* [4]. In this case, sample s and detector d match if a position p exist, where all characters of s and d are identical over the sequence of length r. For $s = \{1\,0\,0\,0\,1\,1\,1\,0\}$ and $r = 4$ the set D_s of all receptors which are able to recognize sample s is

$$D_s = \{0|1000, 1|0001, 2|0011, 3|0111, 4|1110\},$$
$$|D_s| = l - r + 1.$$

Here e.g. the symbol 0|1000 informs that when comparing a string $h \in \mathbb{H}^l$ with such an r-chunk, we should start from 0-th position in h and we must check if 4 contiguous bits of h agree with the template 1000.

4.1.2. Short review of algorithms generating binary receptors and its application

Forrest *et al.* [12] proposed an algorithm generating binary receptors, where candidate detectors were randomly generated; those candidates which recognized any self sample were discarded. This process was repeated until required number of detectors was generated. The algorithm, although highly inefficient, can supplement the performance, for example, of non-random algorithms for a high-dimensional dataset (see below for complexity details).

D'haeseleer *et al.* [11] presented two algorithms, termed *linear time detector generating algorithm* and *greedy detector generating algorithm*, respectively. Both of them try to generate receptors in a more sophisticated way; particularly the greedy algorithm chooses detectors that are far apart, in order to avoid possible overlapping of detectors and to provide enough coverage in the nonself space.

Wierzchoń in [28] used the notion of *binary template* (introduced in previous subsection) to generate "optimal" repertoire by choosing the detectors recognizing as much as possible nonself strings. Using this algorithm, it is possible to find *in advance* the number of detectors needed to cover the set N and to determine the number of holes.

These algorithms use the r-contiguous matching rule. For the r-chunk matching rule, Stibor [23] proposed an algorithm called *Build-Rchunk-Detectors* generating all possible r-chunk detectors, which will not cover any element in S. This algorithm is also appropriate for higher alphabets.

All algorithms mentioned above, except random one, has runtime and space complexity exponential in r, i.e. $O(|\Sigma|^r)$, where $|\Sigma|$ stands for the cardinality of the alphabet used to construct strings of interest. Therefore, they are only applicable to small values of r. Even for the binary alphabet

$|\Sigma| = 2$ and larger values of r the space and time complexity are infeasible high [22]. On other hand, for too small values of r it is not possible to generate any detectors.

Furthermore, Stibor in [23] showed that there is a coherence between the number of generable detectors and the number of resulting holes T,

$$T = \left[2 - \frac{1}{2}\left(1 - \frac{1}{2^{r+1}}\right)^{4|S|} - 2\left(\left(1 - \frac{1}{2^{r+1}}\right)^{2|S|} - \left(1 - \frac{1}{2^{r+1}}\right)^{3|S|}\right)\right]. \tag{3}$$

This means that to minimize the number of holes r must be close to l, but it involves the increase in the complexity of algorithms generating receptors.

Hofmeyr in [15] developed immune-based IDS using binary representation and r-contiguous rule as a matching function. Detectors, were randomly generated and each network TCP SYN packet was encoded as a binary string of length 49:

- 32 bits for IP address of external host,
- 8 bits for IP address of local host (it was assumed, that all internal host have the same class C IP address; in other case, samples were extended to strings of length 76),
- 8 bits representing the type of service,
- 1 bit for indicating the whether or not the first host is the server.

Unfortunately, those attributes are not sufficient to detect intruders in real-world. Therefore, the list of attributes describing the connections should be extended (compare rules used by Snort and number of attributes in KDD Cup dataset) and this involve the use of much longer binary strings.

4.2. Real-Valued representation

4.2.1. V-Detector algorithm

To overcome scaling problems inherent in Hamming space, Ji and Dasgupta [16] proposed real-valued negative selection algorithm, termed as *V-Detector*.

It operates on (normalized) vectors of real-valued attributes; each vector can be viewed as a point in the d-dimensional unit hypercube, $U = [0,1]^d$. Each self sample, $s_i \in S$, is represented as a hypersphere centered at $c_i \in U$ and constant radius r_s, i.e. $s_i = (c_i, r_s)$, $i = 1, \ldots, l$, where l is the number of self samples. Every point $u \in U$ which lies within any self hypersphere s_i is considered as a self element. Also, detectors d_j are represented as hyperspheres: $d_j = (c_j, r_j)$, $j = 1, \ldots, p$ where p is the number of detectors. In contrast to self elements, the radius r_j is not fixed but is computed as the Euclidean distance from a randomly chosen center c_j to the nearest self element (this distance must be greater than r_s, otherwise detector is not created). Formally, we define r_j as

$$r_j = \min_{1 \le i \le l} dist(c_j, c_i) - r_s. \tag{4}$$

The algorithm terminates if predefined number p_{max} of detectors is generated or the space $U \backslash S$ is sufficiently well covered by these detectors; the degree of coverage is described by the parameter co — see [16] for the algorithm and its parameters description.

4.2.2. Selected improvements of V-Detector algorithm

The results presented in [26] show that the original V-Detector algorithm is able to detect only about 2–3% of anomalies in some parts of testing KDD Cup 1999 dataset while statistical anomaly detection techniques recognized more than 80%. It means that V-Detector needs radical improvements.

Below selected improvements of the original algorithm are described (see [10] for details).

1. **Hierarchical decomposition of testing datasets.** It relies upon splitting the dataset according to *a priori* selected (symbolic) attributes, for example: *protocol*, *service*, *flag*, etc. Hence, each subset contains less samples what results in reduction of learning process duration. Besides, if some subsets have attributes, with values identical for all samples, then the problem space is reduced. For example, dimensionality of KDD Cup 1999 dataset can be reduced to 20-30. More details and advantages of such a decomposition is described in [8].

2. **Spanning the problem space over self samples only.** Probably the main reason why poor results were obtained for multidimensional datasets is that the problem space (hypercube) was determined by both the self and nonself samples. Alternatively [6], detectors can be generated only inside the space spanned over self samples. As a consequence, all samples lying outside this space will be classified as nonself, see Fig. 1(b). This allows significantly improve the efficiency of the algorithm; it was possible to classify correctly over 95% (and for many subsets even 100%) samples. This result was much better than that one obtained by using Support Vector Machine — very strong classification tool (see [7] for details).

3. **Use of non-Euclidean metrics.** In its original version, the V-Detector algorithm employs Euclidean distance to measure proximity between two samples. Therefore, self samples and the detectors are hyperspheres (see Figure 1). Formally, Euclidean distance is a special case of Minkowski norm L_m, where $m \geq 1$, which is defined as

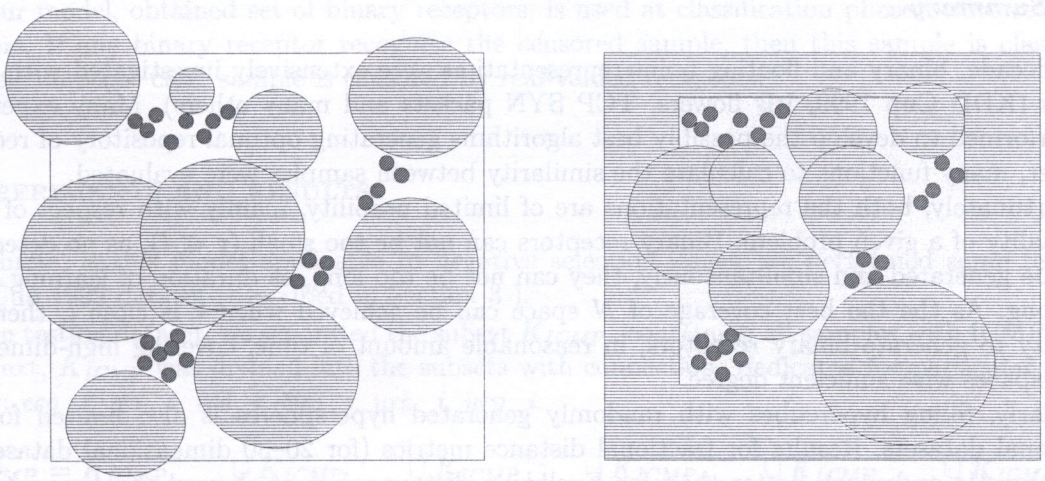
$$L_m(\mathrm{x}, \mathrm{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^m \right)^{\frac{1}{m}} \tag{5}$$

where $\mathrm{x} = (x_1, x_2, \ldots, x_d)$ and $\mathrm{y} = (y_1, y_2, \ldots, y_d)$ are points in $\Re^d$. Particularly, L_2-norm is Euclidean distance, L_1-norm is Manhattan distance, etc.

As observed and proved, for example in [1, 5], L_m-norm loose its meaningfulness of proximity distance for high values of m in highly dimensional spaces. In the case of Euclidean distance, this effect is observed for $d > 15$ and this can explain poor results obtained by Stibor [26] for KDD Cup 1999 dataset.

Using Minkowski norm, Aggarwal *et al.* [1] introduced *fractional distance metric* with $0 < m < 1$, arguing that such a choice is more appropriate in highly dimensional spaces. Experiments, reported in [9], confirm partial efficiency of this proposition. For the $L_{0.5}$-norm we detected almost 80% nonself samples, when only norm was changed in original algorithm. Obtained results were even about 30% better, than for the L_2-norm.

Theoretically, according to proofs mentioned by Aggarwal, when value m decreases, norm should be more appropriate for higher dimensions. Unfortunately, the experiments performed with KDD Cup 1999 dataset in [9], showed that for low values of m ($m < 0.5$) the efficiency of the algorithm decreases to value 0 when $m \leq 0.2$. It implies, that the optimal value of m locates somewhere in the interval $[0.5, 1.0]$; hence, for all the datasets, this value should be properly tuned. Figure 2 presents the unit spheres (shapes of self samples and detectors) for selected *fractional* L_m-norms in 2D.



(a) (b)

Fig. 1. Example of performance V-Detector algorithm for 2-dimensional problem. Grey circles denotes self samples, dashed circles denotes V-detectors, dashed area denotes detector which recognize all samples laying outside the space spanned over all self samples and white areas denotes holes; (a) original algorithm, (b) modified

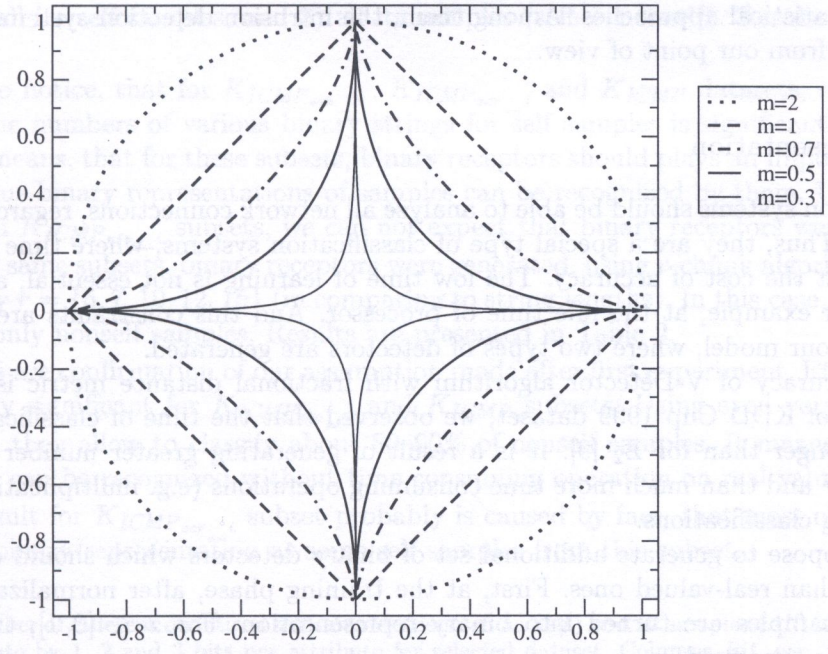


Fig. 2. Unit spheres for selected L_m norms in 2D

The same results showed also a trade-off between efficiency and time complexity for $L_{m<1}$. For fractional norms, the algorithm runs slower for lower m; duration of learning process for $L_{0.5}$ is even 2–3 times higher than for L_2. Similar experiments also were performed by Ji *et al.* [17] for $m \geq 1$. This time, it was observed, that algorithms runs faster and with higher efficiency for lower m with one exception. Namely, algorithms runs faster for L_∞ norm, but unfortunately with the lowest efficiency. Therefore, for applications where the learning or/and classification time is crucial, the norms L_1, L_2 and L_∞ are more appropriate and fractional norms can be used to ensure the higher efficiency, especially for high-dimensional datasets.

4.2.3. Summary

In last decade, binary and floating point representation were extensively investigated with various datasets (KDD Cup 1999, Iris flowers, TCP SYN packets and many others). Many experiments were performed to develop the possibly best algorithms generating optimal repository of receptors. Moreover, many functions to calculate the similarity between samples were evaluated.

Unfortunately, both the representations are of limited usability, mainly with respect of the dimensionality of a given problem. Binary receptors can not be too small ($r \ll l$), as no detectors at all can be generated and simultaneously, they can not be too long, as duration of learning process is too long. As the the best coverage of N space can be achieved when r is close l, there is no possibility to generate binary receptors, in reasonable amount of time, covering high-dimensional domain spaces with sufficient degree.

Similarly, filling hypercubes with randomly generated hyperspheres is also limited for high-dimensional datasets. Results for fractional distance metrics (for 20–30 dimensional datasets) are very optimistic and much better than for Euclidean distance, but as showed the experiments, m can not be infinitely decreased (to value 0). Thus, applying the V-Detector for higher dimensions (e.g. more than 30) can be questionable.

Thus, it is obvious that negative selection algorithms are not applicable, for example, to solving the problem of handwritten digit recognition and many others, where dimensionality is really huge. Since MNIST database [18] contains samples of dimensionality 784 (28×28 fixed-size images), it is clear, that it was not feasible to obtain any classification results in reasonable time [27]. However, in real-world, there are many domains, where presented negative selection algorithms can be efficient alternative for statistical approaches. Among them, the intrusion detection systems, which are the most interesting from our point of view.

4.3. Dual representation

Intrusion detection systems should be able to analyze all network connections, regardless of intensity network traffic. Thus, they are a special type of classification systems, where time of classification is crucial, even at the cost of accuracy. The low time of learning is not essential, as detectors can be generated, for example, at the idle time of processor. And this constraints are taken into the consideration in our model, where two types of detectors are generated.

Although, accuracy of V-Detector algorithm with fractional distance metric is acceptable for selected subsets of KDD Cup 1999 dataset, we observed that the time of classification for $L_{0.5}$ is about 3 times longer than for L_2 [9]. It is a result of generating greater number of detectors to cover subspace N and than much more time consuming operations (e.g. multiplications) have to be performed during classifications.

Hence, we propose to generate additional set of binary detectors which should classify samples in shorter time than real-valued ones. First, at the training phase, after normalization to unitary hypercube, self samples are turned into binary representation. For $x \in [0, 1]$, the quantization function Q can be expressed as

$$Q(x) = \lfloor M * x \rfloor, \tag{6}$$

where $\lfloor \cdot \rfloor$ is floor function and M is the number of ranges (clusters). In our case, the optimal number of clusters is $M = 2^k$, for $k = \{1, 2, \ldots\}$. Assuming that all 41 attributes were quantized to 8 clusters ($k = 3$, 3 bits per each attribute), we will have strings of length equal $l = 41 * k = 123$.

Next, using one of the known algorithm (r-chunk or r-contiguous matching rule), detectors should be generated. Value r is a priori selected and it should be possible low to minimize the time of learning. Simultaneously, r should be sufficiently high to produce possible high amount of detectors as the efficiency of our model increase with the number of detectors. The process of generation binary receptors is terminated when all possible candidates were checked, predefined number of detectors is created or time limit is exceeded.

In our model, obtained set of binary receptors, is used at classification phase before real-valued detectors. If any binary receptor recognize the censored sample, then this sample is classified as a nonself. In other case, sample is censored by real-valued receptors.

5. EXPERIMENTS AND RESULTS

To evaluate, is this model applicable to negative selection issues, we performed some tests with KDD Cup 1999 dataset (described in Section 3).

From testing dataset, we separated the subset K_{ICMP}, containing all samples with ICMP connection. Next, K_{ICMP} was divided into the subsets with connections dedicated for particular services, namely: *eco_i, ecr_i, red_i, tim_i, urh_i, urp_i.*

$$K_{ICMP} = K_{ICMP_{eco\_i}} \cup K_{ICMP_{ecr\_i}} \cup K_{ICMP_{red\_i}} \cup K_{ICMP_{tim\_i}} \cup K_{ICMP_{urh\_i}} \cup K_{ICMP_{urp\_i}}.$$

In all the experiments, training sets were composed from all self samples of particular datasets.

As the $K_{ICMP_{red\_i}}$ and $K_{ICMP_{urh\_i}}$ subsets contains only self samples, we can discard them at the very beginning. All remaining subsets were normalized and subsequently each of 41 attributes were quantized to 2, 4 and 8 equal sized clusters. As a result we obtained binary strings with length 41, 82 and 123, respectively.

First, we determined, for all testing dataset, how many various binary representations can be created for: 1) all its self samples; and 2) all its samples (self + nonself). Results are presented in Table 1.

It is worth to notice, that for $K_{ICMP_{eco\_i}}$, $K_{ICMP_{ecr\_i}}$ and K_{ICMP} datasets, regardless of used quantization, the numbers of various binary strings for self samples is significantly lower than for full subsets. It means, that for these subsets, binary receptors should plays an important part; about 30-60% of various binary representations of samples can be recognized by them. Unfortunately, for $K_{ICMP_{tim\_i}}$ and $K_{ICMP_{urp\_i}}$ subsets, we can not expect that binary receptors were helpful.

Next, for the same subsets, binary receptors were generated, using r-chunk algorithm, for very low r values, namely $r = \{5, 7, 10, 12, 15\}$ (in comparing to string lengths). In this case, the classification sets contained only nonself samples. Results are presented in Table 2.

These results are confirmation of our assumption made after first experiment. Efficiency of binary receptors is very significant for $K_{ICMP_{eco\_i}}$ and K_{ICMP} subsets. Using even very small detectors $r = 5$ or $r = 7$, they allow to classify about 80–90% of nonself samples. It means that, almost all nonself samples can be recognized without time consuming operation on real values.

The poor result for $K_{ICMP_{ecr\_i}}$ subset probably is caused by fact, that most of nonself samples have the same binary representation as some self samples from this subset.

Table 1. Number of different quantized self samples (denoted as *Self*) and all samples (*All*) after quantization of each attribute by 1, 2 and 3 bits per attribute for selected dataset. Columns *Bit. per. [%]* contains the percentage of all quantized samples which are covered by selfs

Dataset	Samples count		Number of different binary representations								
			1 bit per attr.			2 bits per attr.			3 bits per attr.		
	Self	All	Self	All	Bin.rec.[%]	Self	All	Bin.rec.[%]	Self	All	Bin.rec.[%]
$K_{ICMP_{eco\_i}}$	1081	5302	37	61	**39.4**	76	152	**50.0**	152	354	**42.9**
$K_{ICMP_{ecr\_i}}$	3010	3754	16	40	**60.0**	62	108	**42.6**	158	233	**32.2**
$K_{ICMP_{tim\_i}}$	6	9	5	6	**16.6**	6	7	**14.3**	6	7	**14.3**
$K_{ICMP_{urp\_i}}$	2686	2689	45	47	**4.3**	263	266	**1.2**	688	691	**0.4**
K_{ICMP}	4427	11910	54	111	**51.4**	270	406	**33.5**	626	903	**30.7**

Table 2. Number of detectors (*Det.*) and percentage of recognized nonself samples (*Rec.[%]*) for various quantizations (1, 2 and 3 bits per attribute) and r parameters

Dataset	Nonself count	r	1 bit per attr.		2 bit per attr.		3 bit per attr.	
			Det.	Rec.[%]	Det.	Rec.[%]	Det.	Rec.[%]
$K_{ICMP_{eco\_i}}$	4221	5	3	18.57	0	0.00	0	0.00
		7	65	88.60	27	41.55	12	7.08
		10	886	88.89	772	92.42	650	67.78
		12	3908	88.94	3731	92.5	3505	78.65
		15	32549	88.94	32193	93.06	31802	96.04
$K_{ICMP_{ecr\_i}}$	744	5	10	6.53	0	0.00	0	0.00
		7	89	16.67	49	7.34	12	0.27
		10	961	17.88	866	18.01	139	18.68
		12	4018	17.88	3882	18.28	3582	19.35
		15	32669	17.88	32467	20.03	32008	19.62
$K_{ICMP_{tim\_i}}$	3	5	17	0.00	12	0.00	12	0.00
		7	102	0.00	93	0.00	95	0.00
		10	974	0.00	962	0.00	968	0.00
		12	4038	66.67	4015	0.00	4024	0.00
		15	32515	66.67	31550	0.00	29960	0.00
$K_{ICMP_{urp\_i}}$	3	5	1	0.00	0	0.00	0	0.00
		7	55	33.33	7	0.00	0	0.00
		10	866	100.00	596	0.00	315	0.00
		12	3878	100.00	3392	0.00	2654	0.00
		15	32515	100.00	31550	66.67	29960	0.00
K_{ICMP}	7483	5	1	0.07	0	0.00	0	0.00
		7	50	51.40	2	0.04	1	0.00
		10	5878	78.55	3983	53.23	742	9.92
		12	6040	80.72	3284	78.46	2638	51.41
		15	32492	81.95	31331	88.32	29917	69.56

6. Conclusions

This paper present the dual representation model of samples for negative selection issues. We use both types of receptors (binary and real-valued) to minimize time consuming operations on real numbers through replacing them by much faster operations, namely, the comparing of binary strings.

The very preliminary results presented in this paper shows, that even more than 80% nonself samples can be classified by very short binary receptors (in comparing to the sample's length). It is the significant result, with regard the size and dimensionality of testing dataset. However, we realize, that in some cases this model can be useless, like for two datasets presented in this paper. Thus, this model should be in detail investigated with various datasets in the future to confirm its usefulness.

Acknowledgement

This work was partly supported by Technical University of Bialystok grant S/WI/5/03.

REFERENCES

[1] C. Aggarwal, A. Hinneburg, D.A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In: *Proceedings of 8th International Conference on Database Theory*, pp. 420–434, 2001.

[2] U. Aickelin, J. Greensmith, J. Twycross. Immune system approaches to intrusion detection — a review. In: *Proceedings of 3rd International Conference on Artificial Immune Systems*, LNCS Vol. 3239, pp. 316–329. Springer, 2004.

[3] U. Aickelin, J. Twycross, T. Hesketh-Roberts. Rule generalisation in intrusion detection systems using SNORT. In: *International Journal of Electronic Security and Digital Forensics*, **1**(1): 101–116, 2007.

[4] J. Balthrop, F. Esponda, S. Forrest, M. Glickman. Coverage and generalization in an artificial immune system. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, New York, July 9–13, 2002, pp. 3–10.

[5] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft. When is "nearest neighbor" meaningful. LNCS Vol. 1540, pp. 217–235, Springer-Verlag, 1999.

[6] A. Chmielewski, S.T. Wierzchoń. Badanie przydatnosci algorytmu generującego V-Detectory do klasyfikacji wybranych zbiorów. *VI Krajowa Konferencja Inżynieria Wiedzy i Systemy Ekspertowe*, Wrocław, 2006, pp. 13–22.

[7] A. Chmielewski, S.T. Wierzchoń. Comparing real-valued negative selection algorithms for intrusion detection applications. In: *Proceedings of 13th International Multi-Conference on Advanced Computer Systems (ACS 2006)*, Międzyzdroje (Poland), October 18–20, 2006, Vol. I, pp. 387–395.

[8] A. Chmielewski, S.T. Wierzchoń. Experiments with the V-Detector algorithm. *System Science Journal*, 2007 (in print).

[9] A. Chmielewski, S.T. Wierzchoń. On the distance norms for multidimensional dataset in the case of real-valued negative selection application. In: *Zeszyty Naukowe Politechniki Białostockiej*, 2007 (in print).

[10] A. Chmielewski, S.T. Wierzchoń. Simple method of increasing the coverage of nonself region for negative selection algorithms. In: *Proceedings of the 6th Computer Information Systems and Industrial Management Applications (CISIM)*, Ełk (Poland), 2007, pp. 155–160.

[11] P. D'haeseleer, S. Forrest, P. Helman. An immunological approach to change detection: Algorithm, analysis and implications. In: *Proceedings of 1996 Computer Security and Privacy*, Los Alamitos, 1996, pp. 110–119.

[12] S. Forrest, A. Perelson, L. Allen, R. Cherukuri. Self-nonself discrimination in a computer. In: *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Los Alamitos, 1994, pp. 202–212.

[13] P.K. Harmer, P.D. Wiliams, G.H. Gunsch, G.B. Lamont. Artificial immune system architecture for computer security applications. *IEEE Transactions on Evolutionary Computation*, *6*: 252–280, 2002.

[14] S. Hettich, S.D. Bay. KDD Cup 1999 Data, http://kdd.ics.uci.edu

[15] S.A. Hofmeyr, S. Forrest. Architecture for an artificial immune systems. *Evolutionary Computation*, **8**(4): 443–473, 2000.

[16] Z. Ji, D. Dasgupta. Real-valued negative selection algorithm with variable-sized detectors. *Genetic and Evolutionary Computation GECCO-2004*, Part I, LNCS Vol. 3102, pp. 287–298. Seattle, WA, USA, Springer-Verlag, 2004.

[17] Z. Ji, D. Dasgupta. Applicability issues of the real-valued negative selection algorithms. *Genetic and Evolutionary Computation GECCO-2006*, pp. 111–118. Seattle, WA, USA, 2006.

[18] Y. LeCun. The MNIST database of handwritten digits, http://yann.lecun.com/exdb/mnist, 1998.

[19] J.K. Percus, O.E. Percus, A.S. Perelson. Predicting the size of the T-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination. In: *Proceedings of National Academy of Sciences USA (90)*, pp. 1691–1695, 1993.

[20] A. Perelson, D. Weisbuch. Immunology for physicists. *Reviews of Modern Physics*, **69**: 1219–1265, 1977.

[21] SNORT, Intrusion Detection System. http://www.snort.org

[22] T. Stibor. *On the Appropriateness of Negative Selection for Anomaly Detection and Network Intrusion Detection*, PhD thesis. Technical University Darmstadt, 2006.

[23] T. Stibor, K.M. Bayarou, C. Eckert. An investigation of r-chunk detector generation on higher alphabets. In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2004)*, LNSC Vol. 3102, pp. 299–307. Springer-Verlag, 2004.

[24] T. Stibor, P. Mohr, J. Timmis, C. Eckert. Is Negative Selection Appropriate for Anomaly Detection? In: *Proceedings of the ACM SIGEVO Genetic and Evolutionary Computation Conference (GECCO 2005)*, Washington, D.C., June 25–29, 2005, pp. 321–328.

[25] T. Stibor, J. Timmis. Comments on real-valued negative selection vs. real-valued positive selection and one-class SVM. In: *Proceedings of the Congress on Evolutionary Computation (CEC-2007)*, Singapore, September 2007. IEEE Press (accepted for publication).

[26] T. Stibor, J. Timmis, C. Eckert. A comparative study of real-valued negative selection to statistical anomaly detection techniques. In: *Proceedings of the 4th International Conference on Artificial Immune Systems (ICARIS-2005)*, LNCS Vol. 3627, pp. 262–275. Springer-Verlag, 2005.

[27] T. Stibor, J. Timmis, C. Eckert. On the use of hyperspheres in artificial immune systems as antibody recognition regions. In: *Proceedings of the 5th International Conference on Artificial Immune Systems (ICARIS-2006)*, LNCS Vol. 4163, pp. 215–228. Springer-Verlag, 2006.

[28] S.T. Wierzchoń. Generating optimal repertoire of antibody strings in an artificial immune system. *Intelligent Information Systems*, pp. 119–133. Springer-Verlag, 2000.

[29] S.T. Wierzchoń. Deriving concise description of non-self patterns in an artificial immune system. In: L.C. Jain, J. Kacprzyk, eds., *New Learning Paradigms in Soft Computing*, pp. 438–458. Physica-Verlag 2001.

[30] S.T. Wierzchoń. *Sztuczne systemy immunologiczne. Teoria i zastosowania.* Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2001.