# Hybrid Negative Selection Approach for Anomaly Detection

Andrzej Chmielewski[1] and Sławomir T. Wierzchoń[2,3]

[1] Faculty of Computer Science, Białystok University of Technology,
ul. Wiejska 45a, 15-331 Białystok, Poland
a.chmielewski@pb.edu.pl
[2] Institute of Informatics, Gdańsk University,
ul. Wita Stwosza 57, 80-952 Gdańsk, Poland
[3] Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland
stw@ipipan.waw.pl

**Abstract.** This paper describes a *b-v* model which is enhanced version of the negative selection algorithm (*NSA*). In contrast to formerly developed approaches, binary and real-valued detectors are simultaneously used. The reason behind developing this hybrid is our willingness to overcome the scalability problems occuring when only one type of detectors is used. High-dimensional datasets are a great challenge for *NSA*. But the quality of generated detectors, duration of learning stage as well as duration of classification stage need a careful treatment also. Thus, we discuss various versions of the *b-v* model developed to increase its efficiency. Versatility of proposed approach was intensively tested by using popular testbeds concerning domains like computer's security (intruders and spam detection) and recognition of handwritten words.

**Keywords:** Artificial immune system, anomaly detection, multidimensional data.

## 1 Introduction

Natural immune system (NIS) prevents living organism against intruders called *pathogens*. It consists of a number of cells, tissues, and organs that work together to protect the body. The main agents responsible for the adaptive and learning capabilities of the NIS are white blood cells called *lymphocytes*. These differentiate into two primary types: B- and T-lymphocytes called also B- and T-cells for brevity. T-lymphocytes are like the body's military intelligence system, seeking out their targets and sending defenses to lock onto them. Next, B-lymphocytes, destroys detected invaders to protected the body. It is only a very short description of NIS; an unacquainted reader is referred e.g. to [3] for further details.

The mechanisms and procedures developed within NIS were an inspiration for *Artificial Immune Systems* (AIS). Negative selection, clonal selection, idiotypic networks are prominent examples of such mechanisms oriented towards

A. Cortesi et al. (Eds.): CISIM 2012, LNCS 7564, pp. 242–253, 2012.

fast and efficient discrimination between own cells (called *self*) and pathogens (called *nonself*). Only fast and effective response on intruders activity can protect organisms against damaging or even die. It is worth to emphasize that in Nature the total number of various types of pathogens is far greater then $10^{16}$, whereas there are about $10^6$ own cells types only. This discrepancy between the two magnitudes illustrates unusual efficiency of detection mechanisms developed by the NIS. It is expected that employing such ideas in computer algorithms will result in their efficiency in such domains like novelty detection, falsification detection, diagnosis systems, computer security (intruders and spam detection) and many others, where binary classification is sufficient and is required to process an huge amount of data. Exhaustive review of current state in domain of AIS was presented in [16].

In this paper, we focus on negative selection. This mechanism is employed to generate a set of detectors and help protect the body against self-reactive lymphocytes. The lymphocytes start out in the bone marrow and either stay there and mature into B-cells (this process is called *affinity maturation*), or they leave for the thymus gland, where they mature into T-cells in the process of *negative selection*. This process has inspired Forrest *et al.* [7] to formulate so-called *negative selection algorithm* (*NSA*). First, detectors (a counterpart of T-lymphocytes) are generated (usually, in random way). Next, freshly generated detector is added to the set of valid detectors only if does not recognize any *self* element. A nice feature of the *NSA* is that it does not need examples of *nonself* samples (counterpart of *pathogens*) to detect them.

A key problem when applying *NSA* in real-life application seems to be its scalability. For example, to detect spam or intruders at computer networks *NSA* should be able operate on high-dimensional data. Moreover, in such domains the classification process have to be performed online, without significant delays, what makes this task much more difficult to solve. Up to now, neither binary (called $b$-detectors) nor real-valued detectors (called $v$-detectors as they are generated by the $V$-Detector algorithm mentioned in subsection 2.2) were not capable to detect anomalies in satisfactory degree.

To overcome this problem, we propose to use both the types of detectors simultaneously. This hybrid, called $b$-$v$ model, as showed performed experiments, provides much better results in comparing to single detection models as well as in comparing to traditional, statistical approaches, even though only positive *self* examples are required at learning stage. It makes this approach interesting alternative for well known classification algorithms, like SVM, $k$-nearest neighbors, etc.

## 2 Negative Selection Algorithm

The NSA, i.e. the negative selection algorithm, proposed by Forrest *et al.*, [7], is inspired by the process of thymocytes (i.e. young T-lymphocytes) maturation: only those lymphocytes survive which do not recognize any *self* molecules.

Formally, let $\mathcal{U}$ be a universe, i.e. the set of all possible molecules. The subset $\mathcal{S}$ of $\mathcal{U}$ represents collection of all *self* molecules and its complement $\mathcal{N}$ in $\mathcal{U}$

represents all *nonself* molecules. Let $\mathfrak{D} \subset \mathcal{U}$ stands for a set of detectors and let $match(d, u)$ be a function (or a procedure) specifying if a detector $d \in \mathfrak{D}$ recognizes the molecule $u \in \mathcal{U}$. Usually, $match(d, u)$ is modeled by a distance metric or a similarity measure, i.e. we say that $match(d, u) = \texttt{true}$ only if $dist(d, u) \leq \delta$, where *dist* is a distance and $\delta$ is a pre-specified threshold. Various matching function are discussed in [8], [11].

The problem relies upon construction the set $\mathfrak{D}$ in such a way that

$$match(d, u) = \begin{cases} \texttt{false} \text{ if } u \in \mathcal{S} \\ \texttt{true} \;\; \text{ if } u \in \mathcal{N} \end{cases} \tag{1}$$

for any detector $d \in \mathfrak{D}$.

A naive solution to this problem, implied by biological mechanism of negative selection, consists of five steps:

(a) Initialize $\mathfrak{D}$ as empty set, $\mathfrak{D} = \emptyset$.
(b) Generate randomly a detector $d$.
(c) If $math(d, s) = \texttt{false}$ for all $s \in \mathcal{S}$, add $d$ to the set $\mathfrak{D}$.
(d) Repeat steps (b) and (c) until sufficient number of detectors will be generated.

Below the binary and real-valued representations of the problem are described.

### 2.1 Binary Representation

This type of representation was applied by Forrest *et al.* [6] to capture anomalous sequences of system calls in UNIX systems and next to model the system for monitoring TCP SYN packets to detect network traffic anomalies (called LISYS) [9].

In case of binary encoding, the universe $\mathcal{U}$ becomes $l$-dimensional Hamming space, $\mathbb{H}^l = \{0, 1\}^l$, consisting of all binary strings of fixed length $l$:

$$\mathbb{H}^l = \{\underbrace{000...000}_{l}, \underbrace{000...001}_{l}, \ldots, \underbrace{111...111}_{l}\}$$

Hence the size of this space is $2^l$. The most popular matching rules used in this case are:

(a) $r$-contiguous bit rule [6], or
(b) $r$-chunks [2].

Both the rules say that a detector bonds a sample (i.e. data) only when both the strings contain the same substring of length $r$. To detect a sample in case (a), a window of length $r$ $(1 \leq r \leq l)$ is shifted through censored samples of length $l$. In case (b) the detector $t_{i,\mathtt{s}}$ is specified by a substring $\mathtt{s}$ of length $r$ and its position $i$ in the string. Below an example of matching a sample by $r$-detector (left) and $r$-chunk for affinity threshold $r = 3$ is given

$$\overbrace{1\;0\;\mathbf{0}\;\mathbf{0}\;\mathbf{1}\;1\;1\;0}^{l} \qquad \leftarrow \text{sample} \rightarrow \qquad \overbrace{1\;0\;\mathbf{0}\;\mathbf{0}\;\mathbf{1}\;1\;1\;0}^{l}$$

$$0\;1\;\underbrace{\mathbf{0}\;\mathbf{0}\;\mathbf{1}}_{r}\;0\;0\;1 \leftarrow r\text{-detector; } r\text{-chunk} \rightarrow *\;*\;\underbrace{\mathbf{0}\;\mathbf{0}\;\mathbf{1}}_{r}\;*\;*\;*$$

Here it was assumed that irrelevant positions in a string of length $l$ representing the $r$-chunk $t_{3,\mathtt{001}}$ are filled in with the star ($*$) symbol. This way $r$-chunk can be identified with schemata used in genetic algorithms: its order equals $r$ and its defining length is $r-1$. Although a single $r$-detector recognizes much more strings than a single $r$-chunk, this last type of detector allows more accurate coverage of the $\mathcal{N}$ space [2].

Further, the notion of the ball of recognition allows to define "optimal" repertoire $\mathfrak{D}$. Namely it consists of the detectors located in $\mathbb{H}^l$ in such a way that they cover the space $\mathcal{N}$ and their balls of recognition overlap minimally. A solution to such stated problem was given in [17]. To construct the $r$-detectors we split all the *self* strings into the templates represented identically as the $r$-chunks and we construct the detectors by gluing these $r$-chunks that do not belong to the set $\mathcal{S}$. More formally, if $t_{i,\mathtt{s}}$ and $t_{j,\mathtt{w}}$ are two candidate $r$-chunks, we can glue them if both the substrings are identical on $r-1$ positions starting from position $i+1$.

Using such an optimality criterion we come to the conclusion that shortest detectors are more desirable as they are able to detect more samples. However, Stibor [14] showed the coherence between $r$ and $l$ values for various cardinalities of $S$ in terms of the probability of generating detectors, $P_g$. He distinguished three phases:

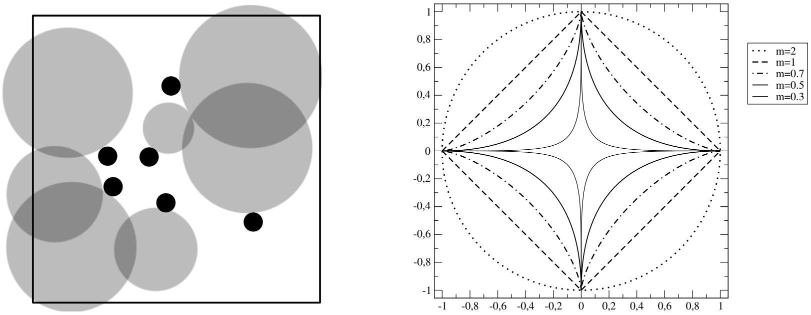
- Phase 1 (for lower $r$) – the probability $P_g$ is near to 0,
- Phase 2 (for middle $r$) – the probability $P_g$ rapidly grows from 0 to 1 (so called *Phase Transition Region*),
- Phase 3 (for higher $r$) – the probability is very near to 1.

Hence, we should be interested in generating detectors with medium length $r$ (belonging to the second region) and eventually with larger values of $r$ if the coverage of $\mathcal{N}$ is not sufficient. It is worth to emphasize, that the detectors can not be too long, due to exponential increase in the duration of learning process, which should be finished in reasonable time.

### 2.2 Real-Valued Representation

To overcome scaling problems inherent in Hamming space, Ji and Dasgupta [10] proposed real-valued negative selection algorithm, termed *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, $\mathcal{U} = [0, 1]^d$. Each *self* sample, $s_i \in \mathcal{S}$, is represented as a hypersphere $s_i = (c_i, r_s)$, $i = 1, \ldots, l$, where



(a) (b)

**Fig. 1.** (a) Example of performance $V$-Detector algorithm for 2-dimensional problem. Black and grey circles denotes *self* samples and $v$-detectors, respectively. (b) Unit spheres for selected $L_m$ norms in 2D.

$l$ is the number of *self* samples, $c_i \in \mathcal{U}$ is the center of $s_i$ and $r_s$ is its radius. It is assumed that $r_s$ is identical for all $s_i$'s. Each point $u \in \mathcal{U}$ inside any *self* hypersphere $s_i$ is considered as a *self* element.

The detectors $d_j$ are represented as hyperspheres also: $d_j = (c_j, r_j)$, $j = 1, \ldots, p$ where $p$ is the number of detectors. In contrast to *self* elements, the radius $r_j$ is not fixed but it 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

$$r_j = \min_{1 \leq i \leq l} dist(c_j, c_i) - r_s \tag{2}$$

The algorithm terminates if predefined number $p_{max}$ of detectors is generated or the space $\mathcal{U} \backslash \mathcal{S}$ is sufficiently well covered by these detectors; the degree of coverage is measured by the parameter *co* – see [10] for the algorithm and its parameters description.

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

$$L_m(\mathrm{x}, \mathrm{y}) = \left( \sum_{i=1}^{d} |x_i - y_i|^m \right)^{\frac{1}{m}}, \tag{3}$$

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, and $L_\infty$ is Tchebyshev distance.

However, Aggarwal *et al.* [1] observed that $L_m$-norm loses its discrimination abilities when the dimension $d$ and the values of $m$ increase. Thus, for example,

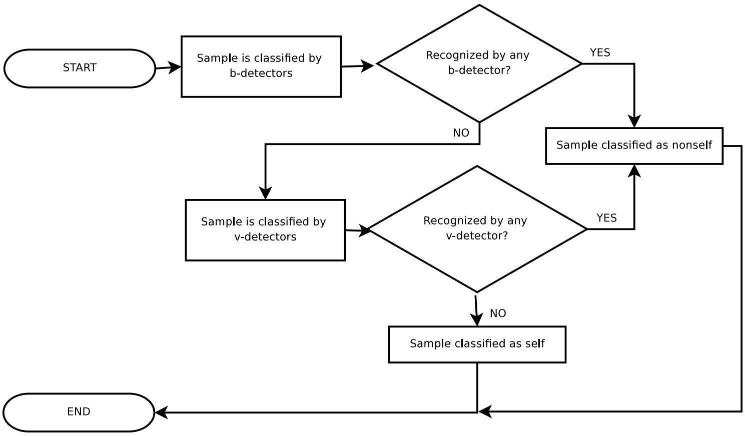

START
Sample is classified by b-detectors
Recognized by any b-detector?
YES
NO
Sample classified as nonself
Sample is classified by v-detectors
Recognized by any v-detector?
YES
NO
Sample classified as self
END

**Fig. 2.** Flow diagram of the classification process for $b$-$v$ model

Euclidean distance is the best (among $L_m$-norms) metrics when $d \leq 5$. For higher dimensions, the metrics with lower $m$ (i.e. Manhattan distance) should be used.

Based on this observation, Aggarwal introduced *fractional distance metrics* with $0 < m < 1$, arguing that such a choice is more appropriate for high-dimensional spaces. Experiments, reported in [4], partially confirmed efficiency of this proposition. For $0.5 < m < 1$, more samples were detected, in comparison to $L_1$ and $L_2$ norms. However, for $m < 0.5$ the efficiency rapidly decreased and for $m = 0.2$, none samples were detected. Moreover, these experiments confirmed also a trade-off between efficiency, time complexity and $m$. For fractional norms, the algorithm runs slower for lower $m$ values; for $L_{0.5}$ the learning phase was even 2-3 times longer than for $L_2$.

Another consequence of applying fractional metrics for $V$-Detector algorithm is modification of the shape of detectors. Figure 1(b) presents the unit spheres for selected $L_m$-norms in 2D with $m = 2$ (outer most), $1, 0.7, 0.5, 0.3$ (inner most).

## 3 Model *b*-*v*

Unsatisfactory coverage of space $\mathcal{N}$ is the main flaw of the $v$-detectors. To overcome this disadvantage as well as to improve the detection rate (DR for short) and to fasten the classification process, a mixed approach, i.e. $b$-$v$ model was proposed. Its main idea is depicted in Figure 2.

Here the binary detectors, as those providing fast detection, are used for preliminary filtering of samples. The samples which did not activate any of $b$-detectors are censored by $v$-detectors next. It is important to note that we do not expect that $b$-detectors covers the space $\mathcal{N}$ in sufficient degree, as it can consume to much time. More important aspect is their length. They should be relatively short (with high generalization degree) to detect as quickly as possible the significant part of *nonself* samples. The optimal length, $r$, of $b$-detectors can be determined by studying the phase transition diagram mentioned in Section 2.1. Namely, we choose the $r$ value guaranteeing reasonable value of the $P_g$ probability what, in addition to ease of generating detectors, results in sufficiently high coverage of the $\mathcal{N}$ space.

In the *b*-*v* model the overall DR ratio as well as the average time of detection depends mainly on the number of recognized *nonself* samples by "fast" *b*-detectors. Thus, in the experiments reported later, we focus mainly on these parameters.

### 3.1 Building *b*-Detectors in Space $\Re^n$

Usually, samples are represented as real-valued vectors. Thus, to construct *b*-detectors, the *self* samples should be converted into binary form first. This can be done in many ways, but probably the simplest one (at least from the computational point of view), is the uniform quantization, [12].

Generally, quantization (used e.g. in digital signal processing, or image processing) refers to the process of approximating a continuous range of values by relatively small set of discrete symbols or integer values. A quantizer can be specified by its input partitions and output levels (called also reproduction points). If the input range is divided into levels of equal spacing, then the quantizer is termed as the uniform quantizer; otherwise it is termed as a non-uniform quantizer, [12].

A uniform quantizer can be described by its lower bound and the number of output levels (or step size). However, in our case, the first of these value is always 0, as we operate only on values from the unit interval (required by the *V*-Detector algorithm). Moreover, for binary representation of output values, instead of the number of output levels, we should rather specify the parameter *bpa*, denoting the number of bits reserved for representing a single level.

The quantization function $Q(x)$ for a scalar real-valued observation $x$, can be expressed as follows:

$$Q(x) = \lfloor x * 2^{bpa} \rfloor, \tag{4}$$

The resulting integer from the range $\{0, 2^{bpa} - 1\}$ is converted to a bit string of length $l = n * bpa$, where $n$ is the dimension of real-valued samples.

### 3.2 Sliding Window for Real-Valued Samples

Experiments conducted on the datasets involving 30-40 attributes have showed, that *V*-Detector algorithm with Euclidean or Manhattan distance metrics provids too low DR values (slightly exceeding 50%-60%, in the best cases). Even the use of fractional distance together with higher values of estimated coverage *co* do not lead to significant improvement of the DR. Note, that due to large number of the attributes involved in the data description it was not possible to construct efficient *b*-detectors.

In practice, e.g. in spam detection or recognition of handwritten letters, the data are characterized by 50 and more (even up to 250) attributes. In such

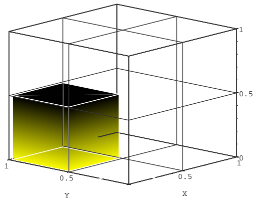


**Fig. 3.** Representation of $b$-detector `101` in space $\Re^3$ for $bpa = 1$

cases, probably none of known metrics is able to measure properly the similarity between the data. Thus, the problem of detecting anomalies becomes very hard (or even impossible) to solve, when only "traditional" methods are used.

Our solution is to incorporate the sliding window idea for the $v$-detectors, to reduce the dimensionality of real-valued samples. This mechanism is already applied for $b$-detectors and is very popular in e.g. segmentation of time series, [5]. Moreover, it is consistent with one of the features of NIS, according to which the *pathogens* are detected by using only partial information [3].

By using sliding window with length $w$, the dimensionality of samples can be reduced from $n$ to $w$. The value of $w$ should be tuned, taking into account the two constraints: (a) it can not be too small as the probability of generating detectors can be too small (similarly to $b$-detectors), (b) it can not be too high as for higher dimensions, still one can meet problem with finding the suitable metric. Hence, the optimal $w$ value seems to be near to the maximal dimensionality for which chosen metric is able to provide satisfactory proximity distance, i.e. for $L_1$ and $L_2$ the most appropriate seems to be $w$ from the range $[5, 20]$.

For example, for $n = 6$ and $w = 4$, each sample vector $\mathtt{x} = (x_1, ..., x_6)$ $(n = 6)$ will be divided on $n - w + 1 = 3$ following parts:

$$x_1, x_2, x_3, x_4;\ x_2, x_3, x_4, x_5;\ x_3, x_4, x_5, x_6.$$

### 3.3 Representation of *b*-Detectors in Space $\Re^n$

$V$-Detector algorithm can take into consideration already generated $b$-detectors, only if they can be represented in space $\Re^n$. In this case, two different shapes of $b$-detectors in real-valued space were investigated: hyperspheres and hypercubes.

The simplest way of converting $b$- to $v$-detector is when $w = r$. Then, the center of $b$-detector ($c_{vb}$) in space $\Re^w$ can be calculated as follow:

$$c_{vb}[k] = \frac{toInt(b_{k*bpa,(k+1)*bpa-1})}{2^{bpa}} + \frac{1}{2^{bpa+1}}, \qquad \text{for } k = 0, \ldots, w-1 \quad (5)$$

where $b_{i,j}$ denotes the substring of $b$-detector from position $i$, to $j$ and $toInt$ is the function which returns the decimal value of the binary number. Depending

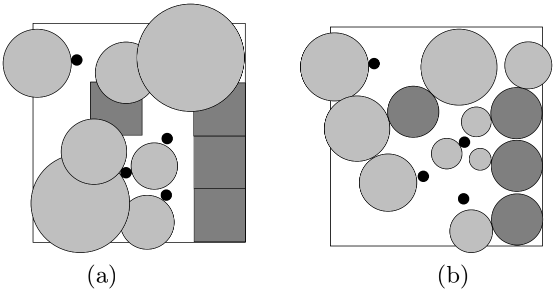


**Fig. 4.** Example of performing *b*-detectors in space $\Re^2$ for $bpa = 2$, when (a) only the centers of both detectors do not cover each other, (b) the subspaces occupied by both detectors are disjoint. *self* samples, *b*- and *v*-detectors are represented as black, dark gray and light gray circles, respectively.

on used shape, the diameter (in case of hyperspheres) or edge (for hypercubes) is equal to $2^{-bpa}$. An example of representation of single *b*-detector in space $\Re^3$ is presented in Figure 3.

### 3.4 Minimizing of Overlapping Regions

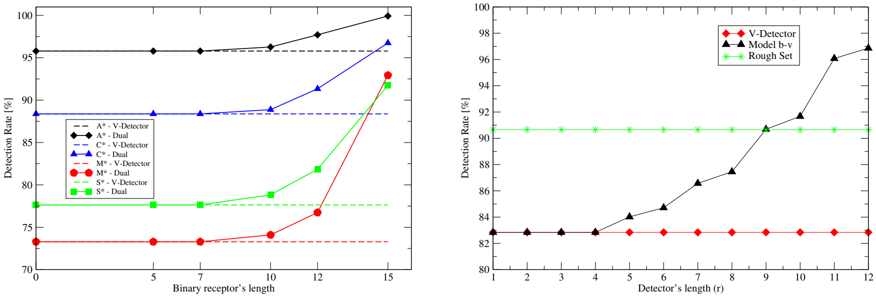
When *b*- and *v*-detectors are generated independently, they could cover the same parts of $\mathcal{U}$. It means, some subset of generated *v*-detectors is superfluous and such detectors should be removed. This way we improve duration of classification which depends on the number of detectors.

Let us denote $\mathfrak{D}_v$ and $\mathfrak{D}_b$ the set of detectors built by $V$-Detector algorithm and those being the real-valued representation of *b*-detectors in space $\Re^n$, respectively. To minimize the overlapping regions, two approaches were considered (see Figure 4). They differentiate according to the shape of $\mathfrak{D}_b$ detectors and allowed overlapping regions.

In Figure 4a, a candidate for *v*-detector is build only when its center is located outside all the detectors from subsets $\mathfrak{D}_v$ and $\mathfrak{D}_b$. In this case the hypercube shape is more appropriate as it is quite easy to check if the detector is covered by any detector $\mathfrak{D}_b$. Moreover, it is acceptable, that some parts of $\mathcal{U}$ can be covered by 2 detectors: one from each set $\mathfrak{D}_v$ and $\mathfrak{D}_b$ (all items from each set are disjoint, by definition).

Other approach is presented in Figure 4b, where detectors from set $\mathfrak{D}_b$ have the same shape as $\mathfrak{D}_v$ detectors. Additionally, the radius of newly generated *v*-detectors were calculated as the distance either to the nearest *self* or *b*-detector. In this way, an overlapping region between $\mathfrak{D}_v$ and $\mathfrak{D}_b$ is empty ($\mathfrak{D}_v \cap \mathfrak{D}_b = \varnothing$).

In both the cases, $V$-Detector algorithm takes into account the $\mathfrak{D}_b$ detectors already generated. As a result, the assumed coverage *co* can be achieved in shorter time. Hence, the overall duration of learning process is faster than in case when both types of detector were generated independently.



(a) (b)

**Fig. 5.** Comparing DR for original $V$-Detector algorithm and $b$-$v$ model: (a) *HWW* dataset, (b) *Diabetes* dataset

## 4 Experiments and Results

The main question we asked ourself, was: *"Is it possible to cover a high-dimensional space $\mathcal{S}$ by both types of detectors, in reasonable time, using various types of similarity metrics?* If the answer would be positive, we can expect that proposed approach is suitable for anomaly detection in high-dimensional problems. Our main goal was to maximize the coverage of $\mathcal{N}$, which should be reflected in DR ratio. Moreover, we were interested also in reducing the classification time, as it is the one of the crucial parameters of on-line classification systems.

Our experiments with $b$-$v$ model were performed on various multidimensional datasets from UCI Machine Learning Repository (*Spambase, Madelon, KDD Cup 1999*) as well as from other popular repositories, including *CBF* (Cylinder-Bell-Funnel, Keogh and Kasetty), *HWW* (*Handwritten Words* [15]) and *Diabetes*[13]. Here, we presents only small part our results.

*HWW* dataset consists of 40 words, each represented by 10 samples. Preliminary experiments has showed, that the recognition of particular words from this dataset is relatively easy task. We took 5 following words: *air*, *name*, *names*, *woman*, *women*. The first word was selected randomly, and next four words have created two pairs of very similar words, which should be rather hard to distinguish. In our experiments, the samples describing one of mentioned word was taken as *self* for which separate sets of detectors were created. Similar approach also was applied for other experiments described in this section.

Fig. 5(a) shows the DR, obtained for the $b$-detectors with the following parameters: $r \in \{3, 4, 5, 7, 10, 12, 15\}$ and $bpa = 1$. Generally, the results agree with the shape of *phase transition region* mentioned at Section 2.1, but here we achieved $DR \approx 100\%$ even for very short detectors (for $r = 12$ or even less, especially, in comparison to the sample length $l = 60$.

Similar experiments were performed for $V$-Detector algorithm with Euclidean and Manhattan distance. Also in this case, all the words were recognized with $DR > 98\%$. Thus we can suppose that this dataset contains easily separated groups of samples, representing particular words. We were not impelled to apply fractional distance metrics, as even Euclidean metric gave highly satisfactory results, although in high-dimensional datasets it should provide the less valuable proximity [1].

The similar DR ratios are showed in Figure 5(b). Our results for $b$-$v$ model are even better than for rough set approach [13], where also *nonself* samples were used at the learning stage.

For all testing dataset, the overall duration of classification was decreased more than 10% in comparing to $V$-Detector algorithm. It is the result of using the very fast $b$-detectors which were able to recognize more than half of censored samples.

## 5 Conclusions

The $b$-$v$-model presented in this paper employs the negative selection mechanism, developed within the domain of Artificial Immune Systems. It is designed for anomaly detection in high-dimensional data, which are difficult to analyze due to the lack of appropriate similarity metrics which enable to cover space $\mathcal{N}$ in sufficient degree and reasonable time. One of the important features of $b$-$v$ model is its ability to minimize the overlapping regions between sets of $b$- and $v$-detectors. As a result the overall duration of classification could be significantly reduced as less $v$-detectors were needed to cover space $\mathcal{N}$. Hence, this model is more efficient for online classification systems in comparing to standard negative selection approaches which based only on one type of detectors.

Moreover, sliding window applied for both types of detectors can be viewed as a possibility to overcome the scaling problem, what makes this model can be applied to solve even the high-dimensional problems, which usually, were beyond the capabilities of *NSA*.

**Acknowledgment.** This work was supported by Bialystok University of Technology grant S/WI/5/08.

## References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the Surprising Behavior of Distance Metrics in High Dimensional Space. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 420–434. Springer, Heidelberg (2001)
2. Balthrop, J., Esponda, F., Forrest, S., Glickman, M.: Coverage and generalization in an artificial immune system. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2002), New York, July 9-13, pp. 3–10 (2002)
3. de Castro, L., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer (2002)

4. Chmielewski, A., Wierzchoń, S.T.: On the distance norms for multidimensional dataset in the case of real-valued negative selection application. Zeszyty Naukowe Politechniki Białostockiej (2), 39–50 (2007)
5. Dasgupta, D., Forrest, S.: Novelty detection in time series data using ideas from immunology. In: Fifth International Conf. on Intelligent Systems, Reno, Nevada, June 19-21 (1996)
6. Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.: A sense of Self for Unix Processes. In: Proc. of the 1996 IEEE Symposium on Research in Security and Privacy, pp. 120–128. IEEE Computer Society Press (1996)
7. 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, pp. 202–212 (1994)
8. Harmer, P.K., Wiliams, P.D., Gunsch, G.H., Lamont, G.B.: Artificial immune system architecture for computer security applications. IEEE Trans. on Evolutionary Computation 6, 252–280 (2002)
9. Hofmeyr, S., Forrest, S.: Architecture for an Artificial Immune System. Evolutionary Computation J. 8(4), 443–473 (2000)
10. Ji, Z., Dasgupta, D.: Real-Valued Negative Selection Algorithm with Variable-Sized Detectors. In: Deb, K., Tari, Z. (eds.) GECCO 2004, Part I. LNCS, vol. 3102, pp. 287–298. Springer, Heidelberg (2004)
11. Ji, Z., Dasgupta, D.: Revisiting negative selection algorithms. Evolutionary Computation 15(2), 223–251 (2007)
12. Sayood, K.: Introduction to Data Compression. Elsevier (2005)
13. Stepaniuk, J.: Rough Set Data Mining of Diabetes Data. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1999. LNCS, vol. 1609, pp. 457–465. Springer, Heidelberg (1999)
14. Stibor, T.: Phase Transition and the Computational Complexity of Generating $r$-Contiguous Detectors. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 142–155. Springer, Heidelberg (2007)
15. Tabedzki, M., Rybnik, M., Saaeed, K.: Method for handwritten word recognition without segmentation. Polish J. of Environmental Studies 17, 47–52 (2008)
16. Timmis, J., Hone, A., Stibor, T., Clark, E.: Theoretical advances in artifical immune systems. Theoretical Computer Science 403(1), 11–32 (2008)
17. Wierzchoń, S.T.: Generating optimal repertoire of antibody strings in an artificial immune system. In: Kłopotek, M.A., Michalewicz, M., Wierzchoń, S.T. (eds.) Proc. of the IIS 2000 Symposium on Intelligent Information Systems, Bystra, Poland, June 12-16, pp. 119–133. Springer (2000)
18. Wierzchoń, S.T.: Deriving concise description of non-self patterns in an artificial immune system. In: Jain, L.C., Kacprzyk, J. (eds.) New Learning Paradigm in Soft Comptuning, pp. 438–458. Physica-Verlag (2001)