DOI: 10.24427/978-83-66391-96-3_2

# Rozdział 2
# Tolerancyjny algorytm V-Detector

*Andrzej Chmielewski*
*Wydział Informatyki, Politechnika Białostocka*

**Streszczenie**: V-Detector jest przykładem algorytmu selekcji negatywnej, bazującym na idei zaczerpniętej ze sztucznych systemów immunologicznych. Przede wszystkim jest wykorzystywany do detekcji anomalii w wielowymiarowych zbiorach danych. Aby osiągnąć ten cel generowane są zbiory detektory, zwykle w sposób losowy, wyłącznie na podstawie informacji o tzw. normalnym zachowaniu i żadne przykłady negatywnych obserwacji nie są wymagane. Głównym problemem zidentyfikowanym podczas zastosowania tego algorytmu jest jego skalowalność i duża złożoność obliczeniowa. W artykule przedstawiono nowe podejście do budowy detektorów, inspirowane ideą tolerancyjnych zbiorów przybliżonych. Za pomocą tego algorytmu można wykryć nie tylko elementy odstające, ale również te, które są bardzo podobne do obydwu klas *self* oraz *nonself*, a ich poprawna klasyfikacja wymaga dodatkowej weryfikacji. Częściowo rozwiązuje również problem skalowalności, ponieważ tego rodzaju detektory mają wyższą moc dyskryminacyjną w porównaniu z wersją podstawową.

**Słowa kluczowe**: sztuczne systemy immunologiczne, selekcja negatywna, detekcja anomalii, zbiory przybliżone

## Wprowadzenie

Algorytm selekcji negatywnej (ang. Negative Selection Algorithm – NSA) jest jednym z kilku głównych algorytmów rozwijanych w ramach dziedziny sztucznych systemów immunologicznych, inspirowanych mechanizmami układów odpornościowych (ang. Nature Immune System – NIS). Jego działanie opiera się na mechanizmach kontrolujących proces dojrzewania tymocytów (młodych limfocytów typu T), w którym przeżywają jedynie te, które nie rozpoznają żadnej własnej molekuły (komórki własnej, określanej jako *self*) [2], czego wynikiem jest zbiór detektorów, rozpoznających jedynie patogeny (komórki obce, określane jako *nonself*). Od strony klasyfikacyjnej, interesującą cechą NIS jest brak konieczności dysponowania przykładami zagrożeń w procesie ich rozróżniania od komórek własnych. Oznacza to, że tego rodzaju algorytmy mogą być stosowane do wykrywania nowych, dotąd nienapotkanych patogenów, które mogą stanowić zagrożenie, a zwykle nie są wykrywane przez tradycyjne

systemy oparte o sygnatury. Powyższa właściwość sprawia, że tego rodzaju rozwiązania są bardzo atrakcyjne w wielu zastosowaniach, szczególnie do tworzenia różnego rodzaju systemów bezpieczeństwa komputerowego, w których największe zagrożenie stanowią ataki, których sygnatury nie zostały jeszcze odnotowane.

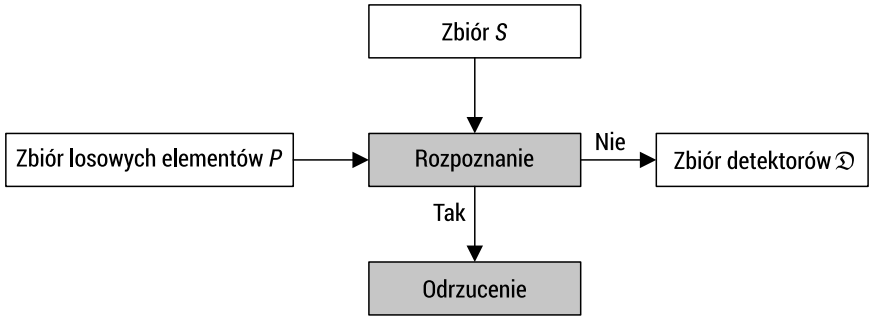
W ostatnich latach zaproponowane zostały liczne algorytmy NSA, m.in. [5, 7, 10], które mają swoje szerokie zastosowanie w wielu obszarach, takich jak choćby bezpieczeństwo komputerowe, filtrowanie spamu, wykrywanie wirusów komputerowych, rozpoznawanie pisma odręcznego (szczegóły m.in. w [6]), w których podstawą jest klasyfikacja binarna. Jednakże, w wielu przypadkach, podział na dwa rozłączne zbiory *self* i *nonself* zwykle nie jest możliwy, ze względu na możliwość występowania elementów o zbyt bliskim podobieństwie do obydwu klas, mierzonym tzw. miarą powinowactwa (odległości pomiędzy nimi). Ewentualna błędna klasyfikacja może pociągać za sobą bardzo negatywne skutki, szczególnie w odniesieniu do systemów bezpieczeństwa. W przypadkach niepewności, takie elementy powinny zostać poddane dodatkowej i bardziej szczegółowej ocenie, również przy użyciu innych technik, często bardziej złożonych czasowo i obliczeniowo od tych oferowanych przez NSA.

Jedną z bardziej efektywnych metod radzenia sobie z klasyfikacją elementów „niepewnych" są zbiory przybliżone (ang. Rough Set Theory – RST) [12]. Pewne próby zastosowania wybranych ich własności w NSA były już podejmowane. Przykładowo w [1] limfocyty, które są odpowiedzialne za wykrywanie i unicestwianie patogenów w NIS, zostały zaimplementowane jako reguły decyzyjne. Zaproponowane rozwiązanie wykorzystywało „przybliżone" limfocyty reprezentowane w postaci wektorów o wartościach rzeczywistych do emulowania niepełnego wiązania dobrze znanego w NIS. W [13] zbiory przybliżone zostały wykorzystane do redukcji liczby atrybutów obiektów w AIS, co bardzo korzystnie wpłynęło na redukcję złożoności obliczeniowej zaproponowanego algorytmu. W tym przypadku testy przeprowadzono na bardzo popularnym zbiorze KDD Cup 1999, będącym wielowymiarowym zbiorem danych, często traktowanym jako referencyjnym, z informacjami zarówno o legalnych połączeniach sieciowych jak i opisami różnego rodzaju ataków.

W tym artykule proponowany jest tolerancyjny algorytm V-Detector inspirowany ideą zaczerpniętą z RST. Wstępne tego rodzaju rozwiązanie było już prezentowane w [3], gdzie wartości dolnej aproksymacji, zarówno w przypadku reprezentacji binarnych ($b$-detektorów), jak i o wartościach rzeczywistych ($\nu$-detektorów), były arbitralnie dobierane w oparciu o wyniki wstępnie przeprowadzonych eksperymentów. W tym przypadku, wartość ta jest uzależniona od liczby elementów znajdujących się w pobliżu badanego obiektu, a przez to automatycznie adoptuje się do lokalnych warunków rozpatrywanego środowiska.

## 2.1. Algorytm selekcji negatywnej

Algorytm selekcji negatywnej (NSA) [2] jest inspirowany procesem dojrzewania limfocytów T (detektorów patogenów) oraz mechanizmami, dzięki którym rozróżniane są antygeny „własne“ (zwane *self* ) od antygenów „obcych” (zwanych *nonself* ). W dużym uproszczeniu, dojrzałe limfocyty charakteryzują się umiejętnością wykrywania komórek obcych i jednoczesnym tolerowaniem komórek własnych, aby nie doszło do reakcji zagrażających życiu gospodarza. Unikalną cechą algorytmu jest wykorzystywanie w procesie generowania detektorów jedynie obserwacji *self*, stąd każdy układ immunologiczny jest unikalny, dopasowany do ochrony konkretnego systemu. Podobnie jak w przypadku układu odpornościowego, kandydaci na detektory są poddawani próbie wykrycia antygenów własnych. Jeśli co najmniej jeden antygen *self* zostanie rozpoznany, musi zostać wyeliminowany. Ogólny schemat tego procesu został przedstawiony na rysunku 2.1.




RYSUNEK 2.1. **Ogólny schemat procesu generowania detektorów $\mathfrak{D}$ za pomocą algorytmu NSA**

W procesie detekcji patogenów (*nonself*), biorą udział jedynie limfocyty dojrzałe, czyli takie, które tolerując wszystkie komórki własne są jednocześnie w stanie związać się jedynie z molekułami komórek patogennych. W idealnym przypadku organizm powinien posiadać tak znaczny repertuar strukturalnie różnych (reagujących na różne molekuły patogenów) limfocytów, aby jak najwięcej patogenów mogło być wykrytych i zneutralizowanych, co jest warunkiem skutecznej ochrony organizmu i zapewnienia mu tym samym bezpieczeństwa.

Formalnie algorytm NSA możemy przedstawić następująco. Oznaczmy zbiór $\mathcal{U}$ jako uniwersum, czyli zbiór wszystkich możliwych molekuł. W jego ramach możemy wyodrębnić dwa rozłączne podzbiory $\mathcal{S}$ oraz $\mathcal{N}$ . Pierwszy z nich $\mathcal{S}$ ($\mathcal{S} \subset \mathcal{U}$ ) zawiera wszystkie molekuły własne (*self*), podczas gdy $\mathcal{N}$ ($\mathcal{N} \subset \mathcal{U}$ ), będący jego dopełnieniem, stanowi przestrzeń dla wszystkich molekuł *nonself*. Przyjmijmy, że $\mathfrak{D} \subset \mathcal{U}$ oznacza zbiór detektorów, a *match*(*d,u*) jest funkcją weryfiującą, czy detektor $d \in \mathfrak{D}$ rozpoznaje molekułę $u \in \mathcal{U}$ . Zwykle *match*(*d,u*) jest modelowane jako metryka odległości lub miara podobieństwa. Wówczas możemy powiedzieć, że *match*(*d,u*) = `true`

tylko w przypadku, gdy $dist(d,u) \leq \delta$, gdzie *dist* jestodległością, a $\delta$ jest predefiniowaną wartością progową, określającą zasięg (moc dyskryminacyjną) detektora. W ostatnich latach były rozpatrywane różnego rodzaju funkcje dopasowywania m.in. w [8] oraz [11], w zależności od wymiarowości zbioru $\mathcal{U}$ oraz liczebności zbioru $\mathcal{S}$.

Od strony formalnej problem sprowadza się do konstrukcji takiego zbioru detektorów $\mathfrak{D}$, aby:

$$match(u,d) = \begin{cases} \texttt{false} \text{ jeżeli } u \in \mathrm{S} \\ \texttt{true} \text{ jeżeli } u \in \mathrm{N} \end{cases} \tag{2.1}$$

dla każdego detektora $d \in \mathfrak{D}$.

Najprostsze rozwiązanie tego problemu, wywodzące się z biologicznych mechanizmów selekcji negatywnej, składa się z następujących etapów:

(a) inicjalizacja $\mathfrak{D}$ jako zbioru pustego, $\mathfrak{D} = \emptyset$,
(b) wygenerowanie losowego detektora $d$,
(c) jeżeli $match(d,s) = \texttt{false}$ dla wszystkich $s \in \mathcal{S}$, dodaj $d$ do zbioru $\mathfrak{D}$,
(d) powtarzaj kroki (b) oraz (c) aż do uzyskania wystarczającej liczby detektorów pokrywających w odpowiednim stopniu przestrzeń $\mathcal{U}$.

Jak dotąd, algorytmy NSA były opracowywane do operowania w przestrzeni ciągów binarnych (*b*-detektorów) oraz liczb rzeczywistych (*v*-detektorów). W każdym z przypadków, rozpatrywane były różne miary podobieństwa dostosowane do danej dziedziny. W następnym rozdziale został przedstawiony algorytm V-Detector operujący w przestrzeni liczb rzeczywistych, który zostanie wyposażony we właściwości RST.

## 2.2. Algorytm V-Detector

Algorytm V-Detector został zaproponowany w [10]. Jest on przystosowany do operowania na obiektach reprezentowanych w postaci wektorów o znormalizowanych wartościach rzeczywistych, które graficznie są przedstawiane w postaci punktu w *d*-wymiarowym jednostkowym hipersześcianie $\mathcal{U} = [0,1]^d$. Każda próbka *self* ($s_i \in \mathcal{S}$) jest w nim reprezentowana jako hipersfera $s_i = (c_i, r_s)$, $i = 1, ..., l$, gdzie $l$ jest liczbą próbek *self*, $c_i \in \mathcal{U}$ jest jej środkiem, a $r_s$ promieniem. Przyjmuje się, że promień $r_s$ jest identyczny dla wszystkich próbek *self*.

Podczas klasyfikacji każdy punkt $u \in \mathcal{U}$ położony wewnątrz dowolnej hipersfery $s_i$ jest traktowany jako obiekt *self*, w pozostałych przypadkach jako *nonself*. Przykład działania algorytmu w przestrzeni 2-wymiarowej został zaprezentowany na rysunku 2.2(a), na którym można zauważyć, że przestrzenie pokrywane przez zbiory $\mathcal{S}$ oraz $\mathfrak{D}$ są rozłączne. W idealnym przypadku, cała przestrzeń $\mathcal{U}$ powinna być pokryta w całości przez *v*-detektory, co jest zadaniem złożonym zarówno pod względem czasowym jak i obliczeniowym.

Do obliczania podobieństwa pomiędzy obiektami zwykle stosowana jest odległość euklidesowa. Konsekwencją tego wyboru jest hipersferyczny kształt detektorów $d_j$: $d_j = (c_j, r_j)$, $j = 1, ..., p$, gdzie $p$ jest liczbą detektorów. W przeciwieństwie do obiektów *self*, promień $r_j$ nie jest wartością stałą i jest wyliczany jako odległość pomiędzy losowo wygenerowanym punktem $c_j$ (nie leżącym wewnątrz żadnej z $s_i$) a najbliższym obiektem *self*. Dodatkowym warunkiem jest to, aby odległość ta była większa niż $r_s$; w przeciwnym przypadku detektor nie jest tworzony. Zapobiega to tworzeniu zbyt małych detektorów (o niskiej mocy dyskryminacyjnej), które mogłyby negatywnie wpłynąć na proces klasyfikacji. Najbardziej pożądane są detektory o dużych promieniach $r_j$, gdyż wówczas mniejsza ich liczba jest potrzebna do pokrycia przestrzeni $\mathcal{U}$.

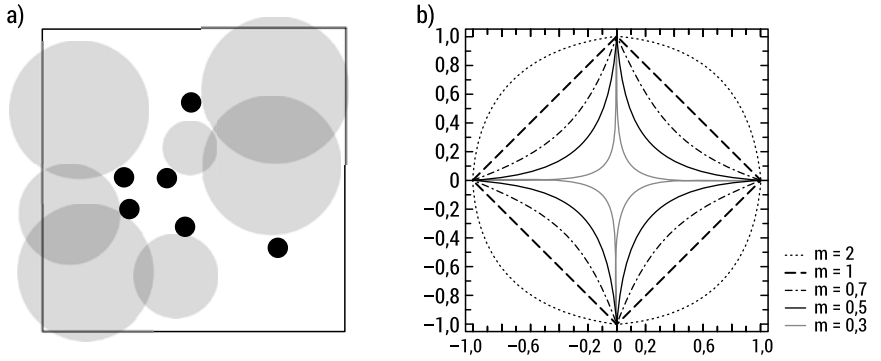
Formalnie $r_j$ można więc zdefiniować jako:

$$r_j = \min_{1 \le i \le l} dist\left(c_j, c_i\right) - r_s . \tag{2.2}$$

W pierwotnie zaproponowanej wersji, algorytm V-Detector wykorzystywał odległość euklidesową do mierzenia podobieństwa pomiędzy obiektami, stąd zarówno obiekty *self* jaki i detektory przybierały kształt hipersfer. Należy jednak mieć na względzie, że metryka euklidesowa jest szczególnym przypadkiem metryki Minkowskiego $L_m$ definiowanej jako:

$$L_m\left(x, y\right) = \left(\sum_{i=1}^{d} \left|x_i - y_i\right|^m\right)^{\frac{1}{m}}, \tag{2.3}$$

gdzie: $x = (x_1, x_2, \ldots x_d)$ oraz $y = (y_1, y_2, \ldots, y_d)$ są punktami w przestrzeni $\Re^d$.




RYSUNEK 2.2. (a) Przykład działania algorytmu V-Detector w przestrzeni 2D. Czarne oraz szare koła oznaczają odpowiednio próbki *self* oraz *v*-detektory, (b) Kształty jednostkowych sfer dla wybranych metryk $L_m$

W szczególności, metryka $L_2$ jest właśnie odległością euklidesową, $L_1$ odległością Manhattan, a $L_\infty$ odległością Czebyszewa. Ponadto, definiowane są również tzw. metryki ułamkowe (dla $0 < m < 1$), które są lepiej przystosowane do operowania na wektorach w przestrzeniach wielowymiarowych [4].

W zależności od zastosowanej w algorytmie V-Detector metryki uzyskuje się inny kształt *v*-detektora. Na rysunku 2.2(b) zostały przedstawione kształty jednostkowych sfer dla wybranych norm $L_m$ w przestrzeni dwuwymiarowej dla $m = 2$ (najbardziej zewnętrzna w postaci okręgu), 1, 0.7, 0.5, oraz 0.3.

## 2.3. Zbiory przybliżone w algorytmach selekcji negatywnej

W ostatnich latach zaproponowanych zostało wiele różnych algorytmów NSA, m.in. [5, 7, 10], które powstały w celu, ogólnie rzecz ujmując, rozwiązania problemu detekcji anomalii (próbek odstających) w wielu dziedzinach np. systemach bezpieczeństwa komputerowego, filtrowania spamu, detekcji wirusów oraz rozpoznawania pisma odręcznego. W celu szerszego spojrzenia na tą tematykę warto zapoznać się choćby z przeglądem zawartym w [6]. Duże zainteresowanie NSA świadczy o potrzebie rozwijania algorytmów, które wykorzystują inne podejście niż te powszechnie stosowane, oparte np. o sygnatury. Kluczową przewagą jest oczywiście fakt możliwości wykrywania nowych typów anomalii bez posiadania informacji o niej.

Jednakże w wielu dotąd opublikowanych artykułach sygnalizowany był problem z niezadowalającą ich skutecznością oraz dużym współczynnikiem błędnie sklasyfikowanych obiektów [14], czego przyczyną było zbyt duże podobieństwo pomiędzy obiektami *self* oraz *nonself*. Próby zastosowania innych miar powinowactwa, często skutkowało poprawą wyników [4], lecz nie było to rozwiązanie uniwersalne.

Jednym z efektywniejszych podejść do danych trudnorozróżnialnych są zbiory przybliżone [9, 12]. Pewne próby inspirowania się tą teorią w odniesieniu do NSA były już publikowane. W [1] limfocyty, które są odpowiedzialne za wyszukiwanie i unicestwianie patogenów w NIS, zostały zaimplementowane jako reguły decyzyjne dla obydwu klas *self* oraz *nonself*. Zaproponowane rozwiązanie wykorzystywało zbiory przybliżone do emulowania niepełnego wiązania znanego z NIS. W [13] zbiory przybliżone zostały użyte do redukcji liczby atrybutów w AIS. W tym przypadku testy były przeprowadzane na referencyjnym zbiorze KDD Cup 1999, wykorzystywanym do oceny detekcji anomalii w ruchu sieciowym.

W dalszej części artykułu zostaną przedstawione podstawowe definicje związane z RST, a następnie tolerancyjny algorytm V-Detector wykorzystujący pewne ich własności. Warto nadmienić, że podobne rozwiązania były już prezentowane w [3], gdzie górna aproksymacja, zarówno w przypadku *b*- oraz *v*-detektorów, była arbitralnie dobierana na podstawie wstępnych eksperymentów przeprowadzanych na podzbiorze danych. W tym przypadku, wartość ta dostosowuje się do bieżących warunków poprzez uwzględnienie pewnej liczby najbliższych obiektów *self*.

## 2.4. Teoria zbiorów przybliżonych

W rozdziale tym zostanie przedstawiony krótki opis, zaczerpnięty z [9], podstawowych zasad wprowadzonych przez koncepcję zbiorów przybliżonych.

Na początek wprowadźmy definicję systemu informacyjnego, na którym oparta jest koncepcja przestrzeni aproksymacyjnej.

**Definicja 1.** (system informacyjny)

System informacyjny jest parą $IS = (U, A)$, gdzie $U$ jest niepustym, skończonym zbiorem obiektów zwanym uniwersum, a $A$ jest niepustym skończonym zbiorem atrybutów. Każdy atrybut $a \in A$ jest traktowany jako funkcja $a$: $U \rightarrow V_a$, gdzie $V_a$ jest wartością zbioru $a$.

**Definicja 2**. (przestrzeń aproksymacyjna)

Parametryczna przestrzeń aproksymacyjna $AS_{\#,\$}$ dla systemu informacyjnego $IS = (U, A)$ jest definiowana jako $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$, gdzie:

- $U$ jest niepustym zbiorem obiektów,
- $I_{\#} : U \rightarrow P(U)$ jest funkcją niepewności,
- $\nu_{\$} : P(U) \times P(U) \rightarrow [0,1]$ jest funkcją inkluzji przybliżonej.

Dla każdego obiektu funkcja niepewności definiuje zbiór podobnie opisanych obiektów.

**Definicja 3**. (funkcja niepewności)

Funkcja niepewności $I_B^{\varepsilon}$ jest definiowana jako:

$$I_B^{\varepsilon}(x) = \bigcap_{a \in B} I_a^{\varepsilon_a}(x), \tag{2.4}$$

gdzie $x \in U, B \subseteq A$, $\varepsilon = (\varepsilon_a : a \in B)$ jest wektorem progów takich, że $\varepsilon_a \geq 0$ dla $a \in B$, $I_B^{\varepsilon}(x) = \{ y \in U : d_a(x,y) \leq \varepsilon_a \}$, a $d_a : U \times U \rightarrow [0, \infty)$ jest miarą odległości.

Funkcja inkluzji przybliżonej określa stopień inkluzji zbioru $X$ w zbiorze $Y$, gdzie $X, Y \subseteq U$.

**Definicja 4**. (funkcja inkluzji przybliżonej)

1. Standardowa inkluzja przybliżona $v_{SRI}(X,Y)$ jest zbiorem $X$ w zbiorze $Y$ jest definiowana jako:

$$V_{SRI} = \begin{cases} \dfrac{card(X \cap Y)}{card(X)} & \text{jeżeli } X \neq \varnothing \\ 1 & \text{w p.p.} \end{cases} \tag{2.5}$$

2. Przybliżona inkluzja $v_{l,u}(X,Y)$ zbioru $X$ w zbiorze $Y$ jest definiowana jako:

$$v_{l,u}(X,Y) = f_{l,u}(v_{\mathrm{SRI}}(X,Y)), \tag{2.6}$$

gdzie:

$$f_{l,u}(t) = \begin{cases} 0 & \text{jeżeli} \quad 0 \le t \le l \\ \dfrac{t-l}{u-l} & \text{jeżeli} \quad l < t < u \\ 1 & \text{jeżeli} \quad t \ge u \end{cases} \tag{2.7}$$

gdzie: $0 \le l < u \le 1$.

Aproksymacja podzbioru $U$ jest definiowana następująco.

**Definicja 5**. (aproksymacja podzbioru uniwersum)

Dolna i górna aproksymacja będąca podzbiorem $X \subseteq U$ w $AS_{\#,\$}$ jest de*fi*niowana jako:

$$LOW(AS_{\#,\$}, X) = \{x \in U : \nu_{\$}(I_{\#}(x), X) = 1\},$$

$$UPP(AS_{\#,\$}, X) = \{x \in U : \nu_{\$}(I_{\#}(x), X) > 0\}.$$

Symbole #,$ oznaczają wektory parametrów, które mogą być dopasowywane w procesie budowania przestrzeni aproksymacji.

**Definicja 6**. (region graniczny)

Region graniczny, który zawiera obiekty niepewne, jest definiowany jako różnica pomiędzy górną i dolną aproksymacją, tj.

$$BND(AS_{\#,\$}, X) = UPP(AS_{\#,\$}, X) \setminus LOW(AS_{\#,\$}, \mathrm{X})$$

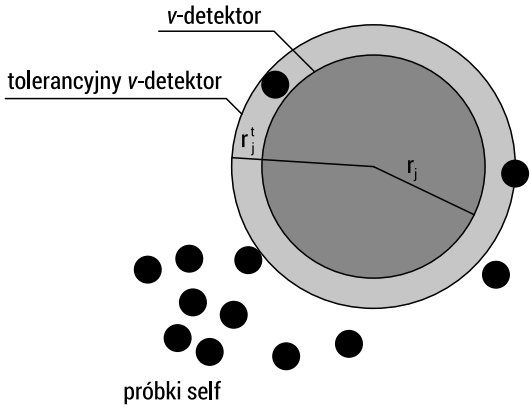
**Definicja 7**. (zbiór przybliżony)

Tolerancyjny zbiór przybliżony podzbioru $X \subseteq U$ jest definiowany jako para $(LOW(AS_{\#,\$}, X), UPP(AS_{\#,\$}, X))$.

## 2.5. Teoria zbiorów przybliżonych

Algorytm V-Detector jest skutecznym narzędziem klasyfikacyjnym szczególnie, gdy przestrzeń, w której operuje, nie jest zbyt wielowymiarowa. Niestety jest przy tym wrażliwy na odstające próbki *self*, które powodują, że znacznie większa liczba detektorów jest potrzebna do pokrycia przestrzeni $\mathcal{U}$, wydłużając w ten sposób proces zarówno uczenia się jak i klasyfikacji.

Zastosowanie zbiorów przybliżonych w odniesieniu do algorytmu V-Detector umożliwia wprowadzenie dodatkowego promienia tolerancji $r^t_j (r^t_j > = r_j)$, który jest odległością do najbliższej próbki *k-self* , zamiast do tej najbliższej. Stąd, każdy tolerancyjny *v*-detektor ($v^t_k$-detektor) w przestrzeni $\Re^d$ jest reprezentowany jako wektor $d^t_j = (c_j, r_j, r^t_j)$ o wartościach rzeczywistych, co graficznie zostało zobrazowane na rysunku 2.3.




RYSUNEK 2.3. **Przykład $v^t_k$-detektora przestrzeni 2D dla $k$ = 2 oraz euklidesowej metryki podobieństwa**

Wprowadzenie dwóch różnych promieni dla pojedynczego detektora $d_j$ zostało zainspirowane przez RST, gdzie $r_j$ jest odpowiedzialny na klasyfikację wszystkich próbek *nonself* z całkowitą pewnością (dolna aproksymacja), podczas gdy $r^t_j$ (górna aproksymacja) jest stosowana do określania obiektów, które mogą również być *nonself* (a przynajmniej są bardzo do nich podobne).

W oparciu o to założenie, podczas fazy klasyfikacji, cenzorowany obiekt jest przypisywany do jednej z 3 grup:

- *nonself* – gdy został całkowicie pokryty przez co najmniej jedną hipersferę z dolną aproksymacją $r_j$;
- *self* – gdy nie został pokryty przez żadną z hipersfer z górną aproksymacją $r^t_j$;
- *uncertain* – w pozostałym przypadku (gdy został pokryty przez co najmniej jedną hipersferę z górną aproksymacją $r^t_j$.

Jak już zostało to wspomniane, V-Detector jest algorytmem dosyć wymagającym pod względem czasowym jak i obliczeniowym. Ta cecha dotyczy obydwu faz tj. uczenia się oraz klasyfikacji. Proponując rozwiązanie przedstawione w tym rozdziale należy również wziąć pod uwagę ewentualny wzrost jego złożoności, warunkujący możliwość zastosowania go w konkretnych dziedzinach.

Ponieważ podczas fazy uczenia się odległość do wszystkich próbek jest zawsze obliczana, wartość $r^t_j$ może być wyznaczona bez żadnego dodatkowego nakładu. Jest to o tyle istotne, gdyż w wielu zastosowaniach (np. w detekcji anomalii w ruchu

sieciowym, wykrywaniu wirusów) szybka adopcja do zaistniałej sytuacji (przebudowanie lub nawet wygenerowanie nowego zbioru $\mathfrak{D}$) jest często jedną z kluczowych własności wszystkich systemów klasyfikacyjnych. Stąd, ewentualny wzrost złożoności mógłby być czynnikiem zmniejszającym jego szanse na zastosowanie.

W fazie klasyfikacji, jedynie w przypadku obiektów *uncertain* jest wymagane przeprowadzenie dodatkowych operacji, które wspomogą ten proces. Ewentualny poziom wzrostu złożoności czasowej i obliczeniowej zależy tu jednak od zastosowanego rozwiązania. Należy jednak zauważyć, że w wielu przypadkach ta operacja niekoniecznie musi być przeprowadzana natychmiast i może być wykonywana np. w czasie bezczynności procesora, aby nie obciążać głównego procesu klasyfikacyjnego. Przykładem takiego rozwiązania, mogą być systemy antywirusowe, czy też systemy detekcji intruzów, które po wykryciu obiektu typu *uncertain*, mogą do czasu ostatecznej klasyfikacji wstrzymywać podejrzane połączenia lub też włączać dodatkowe sensory w celu zebrania bardziej szczegółowych informacji.

## 2.6. Wyniki eksperymentów

Eksperymenty zostały przeprowadzone na dwóch popularnych, wielowymiarowych i o dużej liczności zbiorach danych, czyli KDD Cup 1999 oraz Kyoto 2006+, które zwykle są stosowane do oceny skuteczności algorytmów detekcji intruzów w sieciach komputerowych. W celu zredukowania złożoności obliczeniowej testy były przeprowadzane nie na całych zbiorach, ale na ich częściach zawierających dane dla określonego protokołu transportowego: ICMP, TCP oraz UDP. Dla każdego z nich zostały wygenerowane oddzielne zbiory detektorów. W tym rozdziale zaprezentowane i przeanalizowane zostały wyniki uzyskane dla podzbiorów opisanych w tabeli 2.1, zawierających połączenia na protokole ICMP.

TABELA 2.1. Charakterystyka testowych baz danych wykorzystanych do eksperymentów

| Zbiór danych | Protokół | Liczba atrybutów | Liczba próbek | Liczba próbek *self* |
|---|---|---|---|---|
| KDD Cup 1999 | ICMP | 22 | 11911 | 4428 |
| Kyoto+ | ICMP | 21 | 4069 | 319 |

Ponadto, z obydwu zbiorów danych niektóre atrybuty zostały wyeliminowane (np. adres MAC źródłowy oraz docelowy w przypadku zbioru Kyoto+) ze względu na problemy związane z ich normalizacją (głównie ze względu na zbyt wiele unikalnych wartości symbolicznych). W przypadku zbioru KDD Cup 1999 wszystkie atrybuty niezwiązane z rozpatrywanym protokołem również nie były rozpatrywane, co w konsekwencji pozwoliło na znaczące zmniejszenie ich liczby z 42 do 22. Powyższe operacje pozwoliły na znaczącą redukcję wymiaru przestrzeni $\mathcal{U}$, co w przypadku algorytmu V-Detector ma kluczowe znaczenie w kontekście jego skuteczności i efektywności.

TABELA 2.2. Wyniki uzyskane na zbiorze Kyoto+

| $k$ | $\Delta r$ [%] | *DR* [%] | *FAR* [%] | $\lvert V_t \rvert$ [%] |
|---|---|---|---|---|
| 0 | – | 12,8 | – | – |
| 1 | 11,8 | 30,9 | 4,2 | 16,1 |
| 2 | 17,3 | 39,7 | 7,9 | 24,2 |
| 3 | 19,3 | 43,5 | 8,8 | 26,6 |
| 5 | 26,5 | 59,8 | 13,9 | 45,1 |
| 10 | 36,9 | 73,0 | 23,8 | 54,3 |
| 20 | 51,6 | 87,4 | 34,1 | 67,8 |
| 30 | 58,2 | 93,6 | 41,5 | 73,6 |

TABELA 2.3. Wyniki uzyskane na zbiorze KDD 1999 Cup

| $k$ | $\Delta r$ [%] | *DR* [%] | *FAR* [%] | $\lvert V_t \rvert$ [%] |
|---|---|---|---|---|
| 0 | – | 33,84 | – | – |
| 1 | 3,2 | 48,2 | 0,2 | 6,1 |
| 2 | 6,4 | 61,5 | 0,4 | 16,6 |
| 3 | 9,1 | 48,9 | 0,6 | 32,2 |
| 5 | 10,0 | 58,0 | 0,8 | 34,1 |
| 10 | 18,1 | 59,7 | 1,2 | 34,9 |
| 20 | 18,5 | 77,9 | 3,7 | 42,3 |
| 30 | 21,5 | 78,2 | 4,7 | 80,0 |

Tabele 2.2 oraz 2.3 zawierają wyniki uzyskane odpowiednio dla zbiorów Kyoto+ oraz KDD 1999 Cup. W obydwu przypadkach możemy zaobserwować procentowe wydłużenie długości promienia tolerancyjnych $v$-detektorów ($\Delta r$) wraz z rosnącą wartością $k$, czego rezultatem jest wzrost ich mocy dyskryminacyjnej oraz efektywniejsze pokrycie przestrzeni $\mathcal{U}$. W konsekwencji, w przypadku zbioru KDD Cup 1999, można zaobserwować znaczący wzrost współczynnika wykrycia (*DR*) z 33% dla standardowego algorytmu V-Detector ($k = 0$) do nawet 78% dla $k = 30$. Warty odnotowania jest w tym przypadku dosyć niski współczynnik fałszywego alarmu (*FAR*), co może oznaczać, że nieliczne obiekty *self* były odstające.

Podobne wyniki zostały również uzyskane w przypadku zbioru Kyoto+. O ile wartości $\Delta r$ oraz *DR* rosły znacznie szybciej niż w przypadku drugiego ze zbiorów, to było to niestety powiązane ze znaczącym wzrostem wartości współczynnika FAR, co pokazuje, że wartość $k$ należy dobierać niezwykle ostrożnie, aby zbyt duża liczba obiektów *nonself* nie została błędnie sklasyfikowana jako *self*.

Kolumna $|V_t|$ zawiera informację o procencie próbek sklasyfikowanych jako *uncertain*. Jak można było się tego spodziewać, wartość ta rośnie wraz ze wzrostem liczby $k$, nawet do 80% dla $k = 30$. Dlatego też $k$ nie powinno być zbyt duże, aby w ten sposób zminimalizować liczbę obiektów, które powinny być poddane dalszej, dodatkowej analizie z wykorzystaniem innych algorytmów klasyfikacyjnych.

Wyniki zaprezentowane w tabelach 2.2 i 2.3 uzyskane zostały dla ułamkowej metryki powinowactwa $L_{0.5}$, gdyż wymiarowość testowanych baz danych była zbyt duża, aby można było zastosować odległości euklidesowe oraz Manhattan.

## Podsumowanie

Zastosowanie zbiorów przybliżonych (RST) w *NSA* pozwoliło na określenie dolnej i górnej aproksymacji dla każdego *v*-detektora, zwiększając tym samym jego moc dyskryminacyjną, kosztem utworzenia dodatkowej, oprócz *self* i *nonself*, trzeciej klasy obiektów, nazwanej *uncertain*. Przypisywane są do niej wszystkie cenzorowane obiekty, które są bardzo podobne do obydwu pierwotnych klas, a ostateczna klasyfikacja, w celu uniknięcia błędnego przypisania, wymaga zastosowania innych miar podobieństwa, a nawet algorytmów. Dzięki temu możliwe jest uzyskanie znaczącego wzrostu współczynnika *DR*, przy jednoczesnej redukcji *FAR*.

Warto podkreślić, że wysokie współczynniki *DR* zostały uzyskane bez znaczącego wzrostu złożoności czasowej i obliczeniowej. W rzeczywistości, podczas fazy uczenia się, proces generowania detektorów nie wymaga dodatkowych, obciążających system operacji w porównaniu do pierwotnej wersji algorytmu V-Detector. Natomiast w fazie klasyfikacji, nadprogramowe przetwarzanie dotyczy tylko obiektów zaliczonych do grupy *uncertain*. Pozostałe obiekty są przetwarzane z tą samą złożonością, co w przypadku bazowego algorytmu.

Tolerancyjny V-Detector może znaleźć zastosowanie wszędzie tam, gdzie obiekty *self* oraz *nonself* są słabo rozróżnialne, przez co wiele algorytmów może błędnie je sklasyfikować. Przykładem mogą być różnego rodzaju systemy bezpieczeństwa np. programy antywirusowe oraz systemy detekcji intruzów, w których z pozoru mało znaczące mutacje, w odniesieniu do normalnego zachowania, są często wykorzystywane do skompromitowania celu ataku.

Przyszłe prace zostaną ukierunkowane na m.in. zastosowanie RST w modelu b-v [5] z uwzględnieniem różnych miar podobieństwa. Zasadne wydaje się również zastosowanie innych metryk w procesie pierwotnej klasyfikacji, a innych do cenzorowania obiektów ze zbioru *uncertain*.

## Bibliografia


[1] Acuña, R. F., Ushio, T., *Rough lymphocytes for approximate binding in artificial immune systems*. 15th International Conference on Electronics, Communications, and Computers (CONIELECOMP), 2005, 272–277

[2] Cherukuri, R., Allen, L, Perelson, A. S., Forrest, S., *Self-nonself discrimination in a computer*. Proceedings IEEE Symposium on Security and Privacy, 1994,202–212

[3] Chmielewski, A., *Application of rough sets to negative selection algorithms*. Future Data and Security Engineering – 4th International Conference, (FDSE), 2017, LNCS-10646, 381–394

[4] Chmielewski, A., Wierzchon, S., *On the distance norms for multidimensional dataset in the case of real-valued negative selection application*. „Zeszyty Naukowe Politechniki Białostockiej", 2007, 2, 39–50

[5] Chmielewski, A., Wierzchon,′ S. T., *Hybrid Negative Selection Approach for Anomaly Detection*. 11th International Conference on Computer Information Systems and Industrial Management (CISIM), LNCS-7564, 2012, 242–253

[6] Fernandes, D. A. B. F., Inácio, P., Freire, M., Fazendeiro, P., *Applications of artificial immune systems to computer security: A survey*. „Journal of Information Security and Applications", 2017, 35, 138–159

[7] Forrest, S., Hofmeyr, S. A., Somayaji, A., Longstaff, T. A., *A sense of self for unix processes*. Proceedings IEEE Symposium on Security and Privacy, 1996,120–128

[8] Harmer, P. K., Williams, P. D., Gunsch, G. H., Lamont, G. B., *An artificial immune system architecture for computer security applications*. IEEE transactions on evolutionary computation, 2002, 6(3), 252–280

[9] Honko, P., *Compound approximation spaces for relational data*. International Journal of Approximate Reasoning, 2016, 71, 89–111

[10] Ji, Z., Dasgupta, D., *Real-valued negative selection algorithm with variable-sized detectors*. Genetic and Evolutionary Computation – GECCO 2004, LNCS-3102, 2004, 287–298

[11] Ji, Z., Dasgupta, D., *Revisiting negative selection algorithms*. Evolu-tionary Computation, 2007, 15(2), 223–251

[12] Pawlak, Z,. *Rough sets – theoretical aspects of reasoning about data* (Vol. 9). 1991, Kluwer

[13] Shen, J., Wang, J., Ai, H., *An improved artificial immune system-based network intrusion detection by using rough set*. Communications and Network, 2012, 4, 41–47

[14] Stibor, T., Mohr, P. H., Timmis, J., Eckert, C., *Is negative selection appropriate for anomaly detection?* Genetic and Evolutionary Compu-tation Conference, GECCO 2005, Proceedings, Washington DC, USA, 2005, 321–328


## Tolerant V-Detector algorithm


**Abstract**: V-Detector algorithm is an example of negative selection algorithm, proposed to distinguish between self and nonself samples, represented as real-valued vectors. It is used mainly for detecting anomalies in high dimensional datasets. For this purpose, the set of detectors is generated, usually in random way, base only on information about normal behaviour. The main problem with this algorithm is scalability and high computational complexity. This paper presents a new

approach for building the detectors, inspired by idea from the tolerance rough sets. With this algorithm it is possible to detect an outliers as well as those samples which are very similar to both classes and their correct classification requires additional verification. Partially, it also solves the scalability problem as detectors has a higher discrimination power in comparing to its basic version.

**Keywords**: artificial immune systems, negative selection, rough sets