

NÁVRH ALGORITMŮ NA BÁZI UMĚLÝCH IMUNITNÍCH SYSTÉMŮ PRO ÚLOHY KLASIFIKACE

Luděk Kopáček, Vladimír Olej

Univerzita Pardubice, Fakulta ekonomicko-správní, Ústav systémového inženýrství a informatiky

Abstract: *The paper presents artificial immune systems as a novel approach to classification task. The work is focused on algorithm Immunos-99, CLONALG and CLONCLAS. For algorithm Immunos-99 is designed new version with parameterized dynamical thresholding. These algorithms are used for municipal creditworthiness modelling and benchmark dataset "Iris plant". Finally the comparison with results of another algorithm is presented.*

Keywords: *Artificial immune system, Immunos-99, CLONALG, CLONCLAS, municipal creditworthiness modelling.*

1. Úvod

Umělé imunitní systémy (Artificial Immune Systems, AIS) představují novou oblast výpočtové inteligence. Tyto systémy jsou inspirovány principy a mechanismy přirozených imunitních systémů. Jejich základním úkolem je rozlišovat mezi buňkami vlastního organismu a cizorodými materiály. Látky, které je imunitní systém schopen rozeznat, se nazývají antigeny. Pro umělé imunitní systémy se využívá principů tzv. specifické imunity, která se vyvíjí na základě zkušeností organismu s cizorodými látkami (tzv. naučená imunita). Základními aktéry specifické imunity jsou T-lymfocyty, B-lymfocyty a protilátky. Pro umělé imunitní systémy se často nerozlišují a používají se pouze protilátky.

Na základě použitých principů se umělé imunitní systémy dělí [9] na systémy založené na populacích, systémy založené na principech imunitní auto-tolerance a systémy založené na teorii imunitních sítí. Teorie imunitních sítí pohlíží na imunitní systém jako na síť efektorových buněk, které vzájemně interagují. Imunitní auto-tolerance definuje mechanismy, které zajišťují, že imunitní systém nepůsobí proti buňkám vlastního organismu. Umělé imunitní systémy založené na populacích využívají principů klonální selekce a afinitní maturace pro vývoj populace protilátek.

Pro úlohu klasifikace jsou využívány umělé imunitní systémy založené na populacích, které se během generací vyvíjí. Vývoj této populace prvků je řízen afinitou mezi protilátkami a antigeny, které reprezentují vnější prostředí. Nejvíce stimulované protilátky (s nejvyšší hodnotou afinity) jsou dále klonovány a podstupují proces afinitní maturace, který je také řízen afinitou mezi příslušnou protilátkou a prezentovaným antigenem. Vývoj populace je dán způsobem reprezentace jednotlivých prvků a jejich vzájemného působení prostřednictvím afinity.

Prvním představitelem umělých imunitních systémů, založeným na principech klonální selekce, použitým pro úlohu klasifikace je algoritmus CLONALG (Clonal Selection Algorithm) [5], [9], [10], [14]. V [31] byla na základě tohoto algoritmu navržena nová verze nazvaná CLONCLAS (Clonal Classification). Dalším představitelem AIS, založeným na populacích, je algoritmus AIRS (Artificial Immune Recognition System) [27]. Tento algoritmu je založen na principu tzv. RB nebo ARB (Recognition Ball nebo Artificial Recognition Ball), které by bylo možné označit jako „oblast rozpoznání“ nebo „umělá oblast rozpoznání“. Další skupinou algoritmů určených pro úlohu klasifikace jsou algoritmy Immunos. Prvním představitelem této skupiny populačně založených algoritmů je

algoritmus Immunos-81 [8], na základě kterého byly v [6] navrženy varianty Immunos-1, Immunos-2 a Immunos-99. V této práci budou použity algoritmy Immunos-99, CLONALG a CLONCLAS na klasifikační úlohu modelování bonity obcí a na klasifikaci standardního datového souboru „Iris plant“.

2. Algoritmus Immunos-99

Algoritmus Immunos-99 [6] patří do skupiny algoritmů Immunos. Prvním představitelem této skupiny populačně založených algoritmů je algoritmus Immunos-81 [8]. Dle autora bylo cílem Immunos-81 vytvořit algoritmus s jednoduchou interní reprezentací, schopností zobecnit vstupní data s nominálními, spojitými proměnnými a schopností pracovat s velkým množstvím vzorů. Mezi vlastnosti tohoto algoritmu lze také zahrnout predikovatelnou dobu učení a možnost průběžného učení. Vzhledem k nedostačujícímu popisu tohoto algoritmu byly v [6] vytvořeny algoritmy Immunos-1 a Immunos-2, které byly založeny na stejných principech s cílem reprodukovat výsledky algoritmu Immunos-81. V [6] byl dále navržen algoritmus Immunos-99, který rozšiřuje vlastnosti předchozích verzí. V článku bude pro úlohu klasifikace použita právě varianta Immunos-99.

Vstupní množina s trénovacími daty se v rámci terminologie používané při popisu AIS označuje jako množina antigenů Ag

$$Ag = \{ \mathbf{ag}_i \mid \mathbf{ag}_i = (ag_{i,1}, ag_{i,2}, \mathbf{K}, ag_{i,j}, \mathbf{K}, ag_{i,l}), \mathbf{ag}_i \in S^l \}, \quad (1)$$

kde \mathbf{ag}_i je i -tý antigen množiny Ag , l je počet atributů popisujících daný antigen a $ag_{i,j}$ je j -tý atribut i -tého antigenu \mathbf{ag}_i . Analogicky k množině antigenů Ag lze definovat množinu protilátek Ab

$$Ab = \{ \mathbf{ab}_i \mid \mathbf{ab}_i = (ab_{i,1}, ab_{i,2}, \mathbf{K}, ab_{i,j}, \mathbf{K}, ab_{i,l}), \mathbf{ab}_i \in S^l \}, \quad (2)$$

kde \mathbf{ab}_i je i -tá protilátka Ab , l je počet atributů popisujících danou protilátku a $ab_{i,j}$ je j -tý atribut protilátky \mathbf{ab}_i . Dále je definována množina tříd Cl jako

$$Cl = \{1, 2, \mathbf{K}, n_{class}\}, \quad (3)$$

kde n_{class} je počet tříd pro danou úlohu. Pro další popis se definuje funkce $class$ jako zobrazení ze stavového prostoru S^l do množiny tříd Cl

$$class: S^l \rightarrow Cl. \quad (4)$$

Funkce $class$ tedy pro každý antigen $\mathbf{ag}_i \in Ag$ nebo protilátku $\mathbf{ab}_i \in Ab$ vrátí třídu $c_i \in Cl$, do které patří. Množiny Ab_i a Ag_i lze pak definovat jako

$$Ab_i = \{ \mathbf{ab}_j \mid \mathbf{ab}_j \in Ab; class(\mathbf{ab}_j) = c_i \}, \quad (5)$$

$$Ag_i = \{ \mathbf{ag}_j \mid \mathbf{ag}_j \in Ag; class(\mathbf{ag}_j) = c_i \}. \quad (6)$$

Popis fáze učení algoritmu Immunos-99 pomocí pseudo-kódu je uveden na obr. 1. Prvním krokem v rámci fáze učení algoritmu Immunos-99 je inicializace (funkce `InitImmunos99` na obr. 1), která spočívá v rozdělení množiny antigenů Ag na jednotlivé podmnožiny Ag_i dle (6). Dále je inicializována množina protilátek Ab tak, že z množiny Ag jsou náhodně vybírány antigeny a pro každý z nich je klonováním vytvořena právě jedna protilátka, která je zařazena do množiny této množiny Ab . Velikost množiny Ab je ve vztahu k velikosti Ag dána vstupním parametrem n_{init} fáze učení tohoto algoritmu

$$|Ab| = \lfloor |Ag| \times n_{init} \rfloor. \quad (7)$$

```

[Abm] := Function Immunos99Train (Ag, ninit, ngen, t)
Begin
  Abi = InitImmunos99(Ag, ninit);
  For gen := 1 to ngen do
    For each Abi do
      For each abj ∈ Abi do
        fitj = countFitness (abj, Ag);
      End for;
      [Abi', npc] = performPruning(fit, Abi, t);
      Abi' = performCloningAndMutation(Abi');
      Abi = insertRandomAntigens(Abi', npc);
    End for;
  End for;
  For each Abi do
    For each abj ∈ Abi do
      fitj = countFinalFitness (abj, Ag);
    End for;
    Abm,i = performPruning(fit, Abi, t);
  End for;
  return Abm;
End;

```

Obr. 1: Popis fáze učení algoritmu Immunos 99 pomocí pseudo-kódu

Množina protilátek Ab je dále také rozdělena na podmnožiny Ab_i dle (5). V každé generaci je pro každou protilátku \mathbf{ab}_j z množiny protilátek Ab_i vypočtena hodnotící metrika, která označuje jak dobře daná protilátka \mathbf{ab}_j rozpoznává antigeny z množiny Ag , které patří do stejné třídy. Na obr. 1 je tento koeficient označen jako fit_j , který je výsledkem volání funkce $\text{countFitness}(\mathbf{ab}_j, Ag)$. Hodnota fit_j pro každou protilátku \mathbf{ab}_j z dané množiny protilátek Ab_i je dána jako

$$fit_j = fitness(\mathbf{ab}_j, Ag) = \frac{correct}{incorrect}, \quad (8)$$

$$correct = \sum_{\substack{\mathbf{ag}_x \in Ag \\ class(\mathbf{ag}_x) = class(\mathbf{ab}_j)}} score(\mathbf{ab}_j, \mathbf{ag}_x), \quad (9)$$

$$incorrect = \sum_{\substack{\mathbf{ag}_x \in Ag \\ class(\mathbf{ag}_x) \neq class(\mathbf{ab}_j)}} score(\mathbf{ab}_j, \mathbf{ag}_x). \quad (10)$$

Hodnota $score(\mathbf{ab}_j, \mathbf{ag}_x)$ se stanoví na základě výpočtu metriky $D(\mathbf{ab}_j, \mathbf{ag}_x)$ mezi protilátkou \mathbf{ab}_j a antigenem \mathbf{ag}_x

$$D(\mathbf{ab}_j, \mathbf{ag}_x) = \sqrt{\sum_{i=1}^l (ab_{j,i} - ag_{x,i})^2}. \quad (11)$$

Hodnota $score(\mathbf{ab}_j, \mathbf{ag}_x)$ je pak dána

$$score(\mathbf{ab}_j, \mathbf{ag}_x) = |Ab_i| - index_j, \quad (12)$$

kde $index_j$ je pořadové číslo protilátky $\mathbf{ab}_j \in Ab_i$ v seznamu setříděném podle hodnoty $D(\mathbf{ab}_j, \mathbf{ag}_x)$ v rostoucím pořadí (číslováno od nuly) a $|Ab_i|$ je počet protilátek v množině Ab_i . Pro protilátku \mathbf{ab}_j s nejmenší hodnotou $D(\mathbf{ab}_j, \mathbf{ag}_x)$ (nejvíce podobný antigen \mathbf{ag}_x a protilátka \mathbf{ab}_j) je hodnota $score(\mathbf{ab}_j, \mathbf{ag}_x) = |Ab_i|$ (největší), pro protilátku \mathbf{ab}_k s největší hodnotou $D(\mathbf{ab}_j, \mathbf{ag}_x)$ (nejméně podobný antigen \mathbf{ag}_x a protilátka \mathbf{ab}_j) je hodnota $score(\mathbf{ab}_j, \mathbf{ag}_x) = 1$. Do výpočtu hodnoty fit_j se tak nepromítá absolutní velikost hodnoty $D(\mathbf{ab}_j, \mathbf{ag}_x)$, ale pouze jejich vzájemný vztah. Výpočtem hodnoty fit_j pro všechny protilátky \mathbf{ab}_j z množiny Ab_i je dán vektor ohodnocení této množiny jako

$$\mathbf{fit} = (fit_1, fit_2, \mathbf{K}, fit_j, \mathbf{K}, fit_{|Ab_i|}). \quad (13)$$

Funkce $performPruning(\mathbf{fit}, Ab_i, t)$ odstraní z množiny protilátek Ab_i nekvalitní jedince. Z množiny Ab_i jsou odstraněny protilátky \mathbf{ab}_x , jejichž hodnota fit_x je menší než definovaný práh t , který je vstupním parametrem fáze učení tohoto algoritmu. Množina Ab_i' je tedy po eliminaci vyjádřena takto

$$Ab_i' = \{\mathbf{ab}_j | \mathbf{ab}_j \in Ab_i; fit_j > t\}. \quad (14)$$

Hodnoty fit_j se při nerovnoměrném rozložení protilátek mezi množinami Ab_i mohou řádově lišit. Proto je absolutní nastavení prahu v tomto případě problematické. Jedna hodnota prahu t může úplně potlačit protilátky v jedné množině Ab_i . Pokud je hodnota $t = -1$, je jako práh použita průměrná hodnota prvků vektoru \mathbf{fit} dané množiny protilátek Ab_i . Množina Ab_i' je v tomto případě po eliminaci vyjádřena následujícím způsobem

$$Ab_i' = \{\mathbf{ab}_j | \mathbf{ab}_j \in Ab_i; fit_j > \min(fit_{avg}, 1)\}, \quad (15)$$

kde fit_{avg} je průměrná hodnota atributů vektoru \mathbf{fit} . Tento způsob, který neumožňuje ovlivnit úroveň potlačení, bude dále nazýván jako dynamické prahování bez parametru. Z tohoto důvodu je v tomto článku navržen nový způsob prahování nazvaný dynamické prahování s parametrem, které využívá výhod obou uvedených způsobů. Množina Ab_i' je pak dána

$$Ab_i' = \{\mathbf{ab}_j | \mathbf{ab}_j \in Ab_i; fit_j > \min(|t| \times fit_{avg}, 1)\}. \quad (16)$$

Dynamické prahování s parametrem umožňuje ovlivnit úroveň potlačení protilátek a zároveň respektuje rozsah hodnot fit_j pro danou množinu Ab_i . Počet potlačených protilátek n_{pc} je dán rozdílem velikostí obou populací

$$n_{pc} = |Ab_i| - |Ab_i'|. \quad (17)$$

Funkce $performCloningAndMutation(Ab_i')$ doplní množinu protilátek o nové protilátky, které byly vytvořeny klonováním a mutací stávající množiny protilátek Ab_i'

$$Ab_i = Ab_i' \cup Ab_{clone}, \quad (18)$$

$$Ab_{clone} = \bigcup_{j=1}^{|Ab_i|} Ab_{clone}(\mathbf{ab}_j), \quad (19)$$

kde $Ab_{clone}(\mathbf{ab}_j)$ je množina identických klonů vytvořených z protilátky \mathbf{ab}_j . Pro každou protilátku $\mathbf{ab}_j \in Ab_i$ je vytvořeno $n_{clone}(\mathbf{ab}_j)$ identických klonů

$$n_{clone}(\mathbf{ab}_j) = \left\lfloor \frac{r(\mathbf{ab}_j)}{\sum_{\mathbf{ab}_k \in Ab_i} r(\mathbf{ab}_k)} |Ag_i| + 0,5 \right\rfloor, \quad (20)$$

$$r(\mathbf{ab}_j) = \frac{index_j}{|Ab_i|}, \quad (21)$$

kde $index_j$ je pořadové číslo $\mathbf{ab}_j \in Ab_i$ v seznamu seřazeném podle hodnoty fit_j v rostoucím pořadí a $|Ab_i|$ je počet protilátek v množině Ab_i . Pro protilátku \mathbf{ab}_j s nejmenší hodnotou fit_j je hodnota $index_j = 1$, pro protilátku \mathbf{ab}_k s největší hodnotou fit_k je hodnota $index_k = |Ab_i|$.

Všechny protilátky \mathbf{ab}_k z množiny $Ab_{clone}(\mathbf{ab}_j)$ (tj. všechny klony \mathbf{ab}_k , které vznikly z protilátky \mathbf{ab}_j) následně podstoupí mutaci. Mutace je nepřímo úměrná hodnotě $r(\mathbf{ab}_j)$ vypočtené dle (21). Mutace probíhá pro každý atribut protilátky $\mathbf{ab}_k \in Ab_{clone}(\mathbf{ab}_j)$ odděleně. Funkce `insertRandomAntigens(Ab_i, n_{pc})` připojí do dané množiny Ab_i náhodně vybrané antigeny z Ag_i jejichž počet je dán hodnotou parametru n_{pc} (17), který definuje počet potlačených protilátek v rámci kroku realizovaného funkcí `performPruning(fit, Ab_i, t)`.

Funkce `countFinalFitness(\mathbf{ab}_j, Ag)` vypočítá hodnotu fitness stejným způsobem jako funkce `countFitness(\mathbf{ab}_j, Ag)`, tj. dle vztahu (8), pouze jiným výpočtem hodnoty $score(\mathbf{ab}_j, \mathbf{ag}_x)$

$$score_{final}(\mathbf{ab}_j, \mathbf{ag}_x) = \begin{cases} 1 & \text{pro } \mathbf{ab}_j = \arg \min_{\mathbf{ab}_k \in Ab_i} (D(\mathbf{ab}_k, \mathbf{ag}_x)) \\ 0 & \text{jinak} \end{cases}. \quad (22)$$

Výsledkem fáze učení algoritmu Immunos-99 je množina paměťových protilátek Ab_m , které představuje vnitřní reprezentaci trénovací množiny a je dána jako

$$Ab_m = \bigcup_{i=1}^{n_{class}} Ab_{m,i}, \quad (23)$$

kde $Ab_{m,i}$ je množina paměťových protilátek odpovídající třídě c_i .

Fáze klasifikace spočívá v nalezení správné třídy c_i pro neznámý antigen \mathbf{ag}_x . Proces přiřazení neznámého antigenu k jedné populaci paměťových protilátek $Ab_{m,i}$ je založen na výpočtu avidity [6] mezi neznámým antigenem a všemi populacemi paměťových protilátek $Ab_{m,i}$ pomocí vztahu

$$avidity_1(\mathbf{ag}_x, Ab_{m,i}) = \frac{|Ab_{m,i}|}{\sum_{\mathbf{ab}_j \in Ab_{m,i}} D(\mathbf{ag}_x, \mathbf{ab}_j)}, \quad (24)$$

kde $|Ab_{m,i}|$ je počet protilátek v množině $Ab_{m,i}$ a $D(\mathbf{ag}_x, \mathbf{ab}_j)$ je metrika stanovená dle (11). Neznámý antigen \mathbf{ag}_x je zařazen do třídy reprezentované populací protilátek $Ab_{m,i}$, pro kterou dosahuje avidita (24) největší hodnoty

$$c_x = \arg \max_i (avidity_1(\mathbf{ag}_x, Ab_{m,i})). \quad (25)$$

V rámci popisu fáze učení algoritmu Immunos-99 byly uvedeny tyto parametry, které ovlivňují chování algoritmu:

- Parametr n_{init} [%] – Definuje počáteční velikost množin protilátek Ab_i . Určuje, kolik procent jedinců ze vstupní množiny bude tvořit počáteční populaci. Rozsah hodnot $n_{init} = (0\%; 100\%)$.
- Parametr n_{gen} – Definuje počet generací pro fázi učení.
- Parametr t – definuje práh pro krok eliminace.

3. CLONALG

Algoritmus CLONALG [5], [9], [10], [14] využívá principů klonální selekce a afinitní maturace. Ucelený historický přehled vývoje algoritmu CLONALG lze nalézt v [4]. V [11] byly publikovány dvě verze základního algoritmu označené jako CLONALG1 a CLONALG2, které se liší způsobem formování populace mezi jednotlivými generacemi. V [28] byl publikován algoritmus CLONALG, kde populace byla zpracovávána paralelně na několika procesorech. Navržený algoritmus byl použit na úlohu rozpoznávání vzorů uvedenou v [10]. Uvedená varianta paralelního algoritmu CLONALG demonstruje, že paralelní techniky lze efektivně použít i pro umělé imunitní systémy.

Popis fáze učení algoritmu CLONALG pomocí pseudo-kódu je uveden na obr. 2. Vstupem fáze učení jsou parametry: množina protilátek Ag , počet generací pro fázi učení n_{gen} , velikost populace protilátek n_{Ab} , velikost populace paměťových protilátek n_m , počet protilátek pro fázi selekce n_s , klonální faktor b , počet náhodně vygenerovaných protilátek pro udržení diverzity populace během učení n_d .

```
[Abm] := Function ClonalgTrain (Ag, ngen, nAb, nm, ns, b, nd)
```

```
Begin
```

```
[Abm, Abr] = initClonalg(Ag, nAb, nm);
```

```
Ab = Abm ∪ Abr;
```

```
For i := 1 to ngen do
```

```
  For each agi ∈ Ag do
```

```
    af = affinity(agi, Ab);
```

```
    Abs = select(Ab, af, ns);
```

```
    C = clone(Abs, □, af);
```

```
    C' = mutateClones(C, af);
```

```
    af' = affinity(agi, C');
```

```
    abc = select(C', af', 1);
```

```
    Abm = insert(Abm, abc);
```

```
    Abr = replace(Abr, nd, af);
```

```
  End for;
```

```
End for;
```

```
  return Abm;
```

```
End;
```

Obr. 2: Popis fáze učení algoritmu CLONALG pomocí pseudo-kódu

V rámci inicializace (funkce `initClonalg` na obr. 2) je náhodně vytvořena množina paměťových protilátek Ab_m a množina ostatních protilátek Ab_r , pro které platí

$$|Ab_m| = n_{Ab}, \quad (26)$$

$$n_r = |Ab_r| = n_{Ab} - n_m. \quad (27)$$

Inicializace množin Ab_m a Ab_r probíhá tak, že tyto množiny mají stejné rozložení protilátek mezi třídami jako je rozložení antigenů mezi třídami v množině Ag . Množina protilátek Ab pro fázi učení je pak dána jako sjednocení množiny paměťových protilátek Ab_m a ostatních protilátek Ab_r .

V rámci každé generace je stanovena afinita mezi aktuálně vybraným antigenem \mathbf{ag}_i a populací protilátek Ab . V [10] je pro výpočet afinity použita metrika dle (11). Výsledkem je vektor \mathbf{af} , kde jednotlivé složky vektoru af_j jsou dány jako

$$\mathbf{af} = (af_1, af_2, \mathbf{K}, af_j, \mathbf{K}, af_{n_{Ab}}), \quad (28)$$

$$af_j = D(\mathbf{ag}_i, \mathbf{ab}_j). \quad (29)$$

Vzhledem ke způsobu stanovení afinity v rámci algoritmu CLONALG je nutné v dalším popisu tohoto algoritmu počítat s tím, že malá hodnota afinity af_j dle (29) znamená velkou míru podobnosti antigenu \mathbf{ag}_i a protilátky \mathbf{ab}_j , a opačně.

Na základě vypočteného vektoru \mathbf{af} je vytvořena množina Ab_s výběrem n_s nejlepších protilátek (s nejmenší hodnotou af_j) z Ab ve vztahu k danému antigenu \mathbf{ag}_i . Pro každou vybranou protilátku $\mathbf{ab}_j \in Ab_s$ jsou vytvořeny klony, jejichž počet je dán vztahem

$$nc_j = \left\lfloor \frac{b \cdot n_{Ab}}{i} \right\rfloor, \quad (30)$$

kde i je index v seznamu protilátek z množiny Ab_s seříděný dle af_i od nejlepší po nejhorší. Pro nejlepší protilátku z množiny Ab_s je tedy vygenerováno nejvíce klonů.

Výsledkem funkce clone (viz obr. 2) je množina všech klonů C , která je následně podstoupena mutaci. Mutace tedy probíhá pro jednotlivé množiny klonů C_j , které odpovídají protilátce \mathbf{ab}_j s hodnotou afinity af_j . Klony, které vznikly z protilátky s malou hodnotou af_j , mají malou míru mutace (jsou pozměněny pouze nepatrně), na rozdíl od klonů vzniknuvších z protilátky s velkou hodnotou af_j , které jsou mutací ovlivněny více. Kvalitní jedinci jsou tedy procesem mutace ovlivněni jen nepatrně, na rozdíl od méně kvalitních jedinců, které mutace změní více. Úroveň mutace je dána hodnotou proměnné α , která je vypočtena na základě hodnoty afinity af_j jako

$$a = \exp \frac{af_{\max}}{af_j}, \quad (31)$$

kde af_j je hodnota afinity odpovídající protilátky \mathbf{ab}_j dle (29), ze které byl mutovaný klon vytvářen a af_{\max} je maximální hodnota afinity pro celou množinu protilátek Ab .

Z výsledné množiny C' se pomocí funkce affinity(\mathbf{ag}_i, C') (viz obr. 2) vypočítá hodnota afinity pro všechny nově vytvořené protilátky $\mathbf{ab}_k' \in C'$. Výsledkem je vektor \mathbf{af}' , kde jednotlivé složky vektoru af_j' představují hodnotu afinity mezi antigenem \mathbf{ag}_i a protilátkou $\mathbf{ab}_k' \in C'$

$$\mathbf{af}' = \left(af_1', af_2', \mathbf{K}, af_j', \mathbf{K}, af_{|C'|} \right), \quad (32)$$

kde hodnota af_j' je dána vztahem (29), kde jako argumenty jsou použity antigen \mathbf{ag}_i a protilátka $\mathbf{ab}_k' \in C'$. Následně je z množiny C' vybrána protilátka \mathbf{ab}_c s nejlepší hodnotou afinity stanovenou v předchozím kroku. Pokud je hodnota afinity protilátky $\mathbf{ab}_c \in C'$ ve vztahu

k antigenu \mathbf{ag}_i lepší než hodnota afinity nejlepší paměťové protilátky $\mathbf{ab}_{\text{best}} \in Ab_m$, dojde k jejímu nahrazení protilátkou \mathbf{ab}_c (funkcí replace na obr. 2).

Na závěr každého cyklu pro antigen \mathbf{ag}_i je v rámci funkce replace (viz obr. 2) vyměněno n_d protilátek z množiny Ab_r s nejhorší hodnotou afinity ve vztahu k antigenu \mathbf{ag}_i za náhodně vygenerované protilátky $\mathbf{ab}_y \in S^l$. Fáze učení je ovlivněna nastavením počtu generací n_{gen} , velikostí populace protilátek n_{Ab} , velikostí populace paměťových protilátek n_m , počtem protilátek pro fázi selekce n_s , klonálním faktorem \square , počtem náhodně vygenerovaných protilátek n_d . Výsledkem je množina paměťových protilátek Ab_m . V rámci klasifikace je neznámý antigen \mathbf{ag}_x zařazen do třídy c_x , která odpovídá paměťové protilátce $\mathbf{ab}_{\text{best}} \in Ab_m$ s nejlepší hodnotou afinity ve vztahu k antigenu \mathbf{ag}_x .

4. CLONCLAS

V [31] byla navržena nová verze algoritmu CLONALG nazvaná CLONCLAS (Clonal Classification). Popis fáze učení algoritmu CLONCLAS pomocí pseudo-kódu je uveden na obr. 3. Vstupem fáze učení jsou stejné parametry jako pro algoritmus CLONALG.

```
[Abm] := Function ClonclasTrain (Ag, ngen, nAb, nm, ns, b, nd)
Begin
  [Abm, Abr] = initClonalg(Ag, nAb, nm);
  Ab = Abm ∪ Abr;
  For each agi ∈ Ag do
    For i := 1 to ngen do
      af = affinity(agi, Ab);
      Abs = select(Ab, af, ns);
      C = clone(Abs, □, af);
      C' = mutateClones(C, af);
      af' = affinity(agi, C');
      abc = select(C', af', 1);
      Abm = insert(Abm, abc);
      Abr = copyPopulation(C');
      Abr = replace(Abr, nd, af);
    End for;
  End for;
  return Abm;
End;
```

Obr. 3: Popis fáze učení algoritmu CLONCLAS pomocí pseudo-kódu

Proti algoritmu CLONALG je v tomto případě změněno pořadí provádění jednotlivých cyklů. V případě algoritmu CLONALG se v rámci každé generace projde celá množina antigenů. V tomto případě se množina antigenů prochází pouze jednou (vnější cyklus), ale každý antigen $\mathbf{ag}_i \in Ag$ interaguje s množinou protilátek Ab v daném počtu generací n_{gen} . Zbývající část obou algoritmů je shodná.

Algoritmus CLONCLAS obsahuje navíc funkci copyPopulation, která realizuje částečnou nebo úplnou náhradu množiny Ab_r protilátkami z množiny C' . Zdali dojde k úplné nebo částečné náhradě se řídí vzájemnou velikostí obou množin:

- Pokud pro velikosti obou množin platí vztah $|C'| > |Ab_r| = n_r$, je množina Ab_r plně nahrazena n_r nejlepšími protilátkami z množiny C' podle hodnoty afinity ve vztahu k aktuálnímu antigenu \mathbf{ag}_i .

- Pokud pro velikosti obou množin platí vztah $|C'| = |Ab_r|$, je množina Ab_r plně nahrazena množinou C' .
- Pokud pro velikosti obou množin platí vztah $|C'| = nc < |Ab_r|$, je v množině Ab_r nahrazeno nc nejhorších protilátek (podle hodnoty afinity ve vztahu k aktuálnímu antigenu ag_i) protilátkami z množiny C' .

Fáze klasifikace a popis parametrů ovlivňujících chování tohoto algoritmu je stejný jako pro algoritmus CLONALG.

5. Návrh experimentů

Vyhodnocení popsaných algoritmů Immunos-99, CLONALG a CLONCLAS bude provedeno na dvou klasifikačních úlohách. První úlohou bude modelování bonity obcí, kde na základě známého ohodnocení trénovací skupiny obcí budou klasifikovány obce z testovací množiny. Třídy představují jednotlivé bonitní ohodnocení. Druhou úlohou bude klasická klasifikační úloha na datovém souboru „Iris plant“.

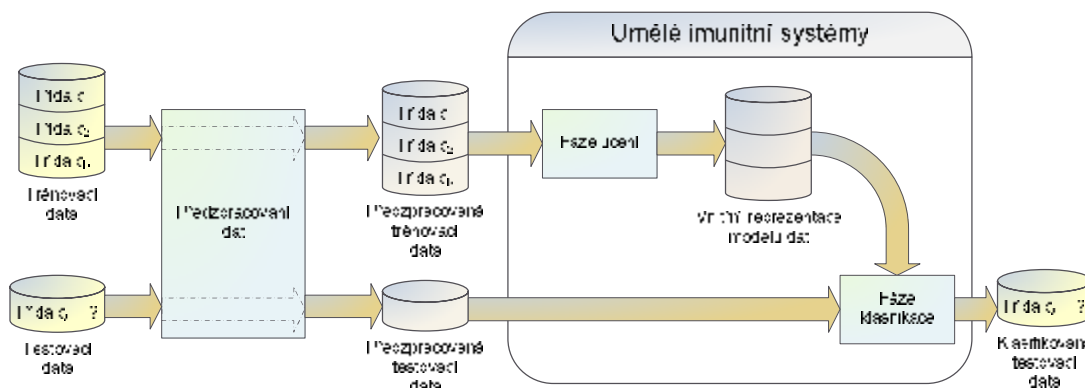
Oba datové soubory budou rozděleny na trénovací a testovací množinu tak, že obě množiny mají stejné rozložení vzorů v jednotlivých třídách [21]. Trénovací data budou použita pro nastavení použitého algoritmu, který si pomocí těchto dat vytvoří vnitřní reprezentaci, kterou následně využije pro klasifikaci neznámých vzorů. Testovací data budou sloužit k ověření klasifikátoru a jeho interní reprezentaci dat. Vzory z trénovací množiny budou ohodnoceny klasifikátorem a výsledná klasifikace bude porovnána se správným přiřazením ke třídě.

Klasifikační úlohu lze definovat jako zobrazení nad dvěma množinami [21]. Každý vzor je popsán vektorem parametrů $\mathbf{x}_i \in X$ a je přiřazen právě do jedné třídy $c_j \in Cl$, kde X je množina vzorů a Cl je množina tříd. Klasifikační pravidlo lze definovat jako zobrazení z množiny vzorů X do množiny tříd Cl

$$I : X \rightarrow Cl, \quad (33)$$

$$c_j = I(\mathbf{x}_i). \quad (34)$$

Zobrazení \square tak představuje rozklad množiny X na n_c vzájemně disjunktčních podmnožin, kde n_c je počet tříd. Cílem klasifikátoru je nalézt na základě trénovací množiny takový rozklad, resp. zobrazení \square' , které nejlépe odpovídá zobrazení \square . Navržený model je znázorněn na obr. 4.



Obr. 4: Modelování bonity obcí pomocí AIS

Vzhledem k náhodné složce algoritmů Immunos-99, CLONALG a CLONCLAS bude v rámci každého experimentu provedeno $n_{pok} = 10$ pokusů pro dané nastavení parametrů. Úspěšnost klasifikace i -tého pokusu Acc_i je dána tvarem

$$Acc_i = \frac{n_{cor,i}}{n_{test}}, \quad (35)$$

kde $n_{cor,i}$ je počet správně klasifikovaných vzorů z testovací množiny a n_{test} je počet vzorů v testovací množině. Každý experiment tak bude vyhodnocen průměrnou úspěšností klasifikace Acc_{avg} a maximální hodnotou úspěšnosti klasifikace Acc_{max} dosaženou na trénovací množině s daným nastavením. Experimenty pro algoritmy Immunos-99 a CLONALG byly realizovány v rámci prostředí WEKA [32] s využitím „Weka Classification Algorithms“ [7].

První úlohou použitou pro vyhodnocení uvedených algoritmů je modelování bonity obcí. Bonita znamená schopnost a ochotu obce splácet svůj dluh. Použitý datový soubor obsahuje hodnocení bonity obcí Pardubického kraje [15], [16], [18], [24] a je tvořen 12 popisnými proměnnými [15], [24], které popisují 452 obcí Pardubického kraje. Hodnoty proměnných odpovídají stavu z roku 2004 a zahrnují dluhové, finanční a demografické ukazatele obcí.

Trénovací množina reprezentuje historické ohodnocení bonity obcí a na jejím základě je vytvořen vnitřní model, podle kterého je stanovena bonita pro nově hodnocenou obec z testovací množiny. Vlastní analýza pro stanovení bonity hodnocené obce je poměrně náročný proces a výsledné hodnocení je finančně náročné. Proto je motivace vytvořit alespoň částečně automatizovanou metodu hodnocení obcí, která vytvoří první náhled na bonitu obce s menší časovou a odbornou náročností a tím i s menšími náklady. V literatuře lze nalézt různé modely na bázi neuronových sítí, fuzzy množin nebo expertních systémů pro predikování a vysvětlení ratingu [2], [3], [22], [23], [25], [33].

Druhým použitým datovým souborem je „Iris plant“ [1], který obsahuje 150 vzorů popisujících jednotlivé kosatce. Každý vzor je popsán pomocí 4 reálných atributů. Jednotlivé vzory jsou rozděleny do tří tříd. Každá třída reprezentuje jeden druh kosatce. Jedna třída je lineárně separabilní od ostatních vzorů, ale zbylé dvě třídy již nejsou lineárně separabilní.

6. Analýza výsledků

Kapitola obsahuje úspěšnosti klasifikace získané jednotlivými algoritmy pro modelování bonity obcí a pro datový soubor „Iris plant“. V tab. 1 je uvedena průměrná a maximální hodnota úspěšnosti klasifikace pro algoritmus Immunos-99 při modelování bonity obcí. Při navrženém způsobu dynamického prahování s parametrem bylo dosaženo lepších výsledků jak z hlediska průměrné úspěšnosti klasifikace, tak i z hlediska jejich maximálních hodnot. Pro dosažení uvedených výsledků bylo při dynamickém prahování s parametrem potřeba menší počet generací, než v případě originální verze bez parametru.

Tab. 1: Výsledky klasifikace pro modelování bonity obcí pomocí algoritmu Immunos-99

Verze alg. Immunos-99	t	n_{gen}	$n_{init}[\%]$	$Acc_{avg}[\%]$	$Acc_{max}[\%]$	\square
dynamické prah. bez parametru	-1	40	90	89.76	90.27	1.06
dynamické prah. s parametrem	-0.9	25	90	90.00	91.15	1.12

Výsledky získané pomocí algoritmů CLONALG a CLONCLAS jsou uvedeny v tab. 2. Algoritmus CLONALG dosáhl vyšší hodnoty maximální úspěšnosti klasifikace $Acc_{max} = 92.04\%$. Na druhou stranu algoritmus CLONCLAS dosáhl větší hodnoty průměrné úspěšnosti klasifikace $Acc_{avg} = 89.20\%$. Nastavení parametrů bylo v obou případech srovnatelné. Pouze v případě algoritmu CLONCLAS byly dosažené výsledky získány s vyšší hodnotou parametru n_s .

Tab. 2: Výsledky klasifikace pro modelování bonity obcí získané algoritmem CLONALG a CLONCLAS

Algoritmus	n_{gen}	n_{Ab}	n_m	n_s	\square	n_d	$Acc_{avg}[\%]$	$Acc_{max}[\%]$	\square
CLONALG	10	425	340	42	0.2	60	88.41	92.04	2.52
CLONCLAS	10	465	372	139	0.1	70	89.20	91.15	1.36

V tab. 3 [17] je uvedeno srovnání dosažených průměrných výsledků algoritmu Immunos-99 (dynamické prahování bez parametru a navržená verze dynamického prahování s parametrem), CLONALG a CLONCLAS s výsledky získanými pomocí jiných algoritmů. Výsledky ostatních uvedených algoritmů byly převzaty z [17]. Z použitých algoritmů byly nejlepší průměrné výsledky úspěšnosti klasifikace dosaženy pomocí algoritmu Immunos-99 s navrženým dynamickým prahováním s parametrem. Použité algoritmy lze použít pro modelování bonity obcí, ale dosažené výsledky otevírají možnost jejich další optimalizace.

Tab. 3: Srovnání nejlepších výsledků získaných v rámci modelování bonity obcí

Algoritmus	$Acc_{avg}[\%]$	$Acc_{max}[\%]$	\square
LVQ1	91.33	92.92	0.97
FFNN	90.56	92.04	1.33
OLVQ1	90.44	92.92	1.45
ARTMAP	90.34	93.36	3.81
LVQ3	90.09	92.04	1.45
Immunos-99 dyn. s param.	90.00	91.15	1.12
RBF	89.93	94.69	2.88
LVQ2	89.91	92.04	1.61
SVM	89.76	91.15	1.83
Immunos-99 dyn. bez param.	89.76	90.27	1.06
CLONCLAS	89.20	91.15	3.03
CLONALG	88.41	92.04	2.52
KNN	87.46	90.27	3.38
LNN	84.60	85.84	0.79
PNN	83.34	85.84	1.89
MLRM	81.42	86.73	5.31

Další experimenty byly provedeny pro datový soubor „Iris plant“. V tab. 4 jsou uvedeny výsledky pro datový soubor „Iris plant“ získané pomocí algoritmu Immunos-99. V rámci provedených experimentů byla úspěšnost klasifikace alespoň jednoho pokusu $Acc_i = 100\%$. V prvním řádku jsou uvedeny výsledky klasifikace pro dynamické prahování bez parametru (tomu odpovídá nastavení $t = -1$). V druhém řádku tab. 4 jsou uvedeny výsledky algoritmu Immunos-99 s navrženým dynamickým prahováním s parametrem. Při tomto způsobu prahování bylo zapotřebí menšího počtu provedených generací. Z uvedených výsledků plyne, že algoritmus Immunos-99 lze úspěšně použít i v případě datového souboru „Iris plant“.

Tab. 4: Výsledky klasifikace pro datový soubor „Iris plant“ pomocí algoritmu Immunos-99

Verze alg. Immunos-99	t	n_{gen}	$n_{init}[\%]$	$Acc_{min}[\%]$	$Acc_{avg}[\%]$	$Acc_{max}[\%]$	\square
dynamické prah. bez parametru	-1	50	70	97.30	98.65	100	1.35
dynamické prah. s parametrem	-0.9	30	80	97.30	98.92	100	1.32

V tab. 5 jsou uvedeny úspěšnosti klasifikace získané pomocí algoritmu CLONALG a CLONCLAS pro datový soubor „Iris plant“. Dosažené výsledky jsou stejné jako v případě algoritmu Immunos-99 s navrženým dynamickým prahováním s parametrem.

Tab. 5: Výsledky klasifikace pro datový soubor „Iris plant“ pomocí algoritmů CLONALG a CLONCLAS

Algoritmus	n_{gen}	n_{Ab}	n_m	n_s	\square	n_d	$Acc_{min}[\%]$	$Acc_{avg}[\%]$	$Acc_{max}[\%]$	\square
CLONALG	10	175	123	35	0.1	2	97.30	98.92	100	1.32
CLONCLAS	10	165	132	49	0.2	1	97.30	98.92	100	1.32

V tab. 6 je uvedeno srovnání dosažených výsledků pro datový soubor „Iris plant“ s jinými algoritmy. Tabulka obsahuje dosažené výsledky, metodu vyhodnocení a odkaz na zdrojovou literaturu. Úspěšnost klasifikace v závorce znamená maximální úspěšnost klasifikace v rámci provedených experimentů. Jak je z této tabulky patrné, tak použité algoritmy Immunos-99, CLONALG a CLONCLAS dosahují ve srovnání s ostatními velmi dobrých výsledků.

Tab. 6: Srovnání dosažených výsledků pro datový soubor "Iris plant"

Algoritmus	$Acc[\%]$	Metoda vyhodnocení	Zdroj
Genetické algoritmy + ANN	100	Vyhodnocení na trénovací množině.	[12]
Grobian	100	Vyhodnocení na trénovací množině.	[12]
CLONALG, CLONCLAS	98.92 (100)	Rozdělení na testovací a trénovací množinu.	viz výše
Immunos-99 dynamické prahování s parametrem	98.92 (100)	Rozdělení na testovací a trénovací množinu.	viz výše
Immunos-99 dynamické prahování bez parametru	98.65 (100)	Rozdělení na testovací a trénovací množinu.	viz výše
Clustering GP (k=3)	97.9	Křížová validace (10 násobná)	[13]
AIRS1	96.7	Křížová validace (10 násobná)	[29]
ANN (BP)	96.7	Metoda „Leave one out“	[26]
AIRS2	96.0	Křížová validace (10 násobná)	[30]
DAIS	95.8	Křížová validace (10 násobná)	[19]
Naive Bayes, Bagging NB	95.53	Křížová validace (10 násobná)	[20]
Bagged C4.5	95	Křížová validace (10 násobná)	[13]
Logitboost NB	94.87	Křížová validace (10 násobná)	[20]
C4.5	94.1	Křížová validace (10 násobná)	[13]

7. Závěr

V [9] lze nalézt aplikace umělých imunitních systémů na úlohy optimalizace, datové analýzy, plánování, apod. Tento článek ukazuje možnost využití vybraných zástupců umělých imunitních systémů pro úlohu klasifikace. Jedná se o algoritmy Immunos-99, CLONALG a CLONCLAS. Pro algoritmus Immunos-99 byla v tomto článku navržena verze dynamického prahování s parametrem, která využívá výhod dynamického prahování a zároveň umožňuje v rámci nastavení parametru ovlivnit úroveň potlačení protilátek ve fázi učení. Uvedené algoritmy včetně navržené varianty algoritmu Immunos-99 byly použity pro úlohu modelování bonity obcí a pro klasifikaci standardního datového souboru „Iris plant“. Pro úlohu modelování bonity obcí, dosahuje algoritmus Immunos-99 ve srovnání s algoritmy CLONALG a CLONCLAS lepších průměrných hodnot úspěšnosti klasifikace Acc_{avg} . Pro úlohu klasifikace datového souboru „Iris plant“ se průměrné hodnoty klasifikace použitých algoritmů liší pouze o $\square Acc = 0.27\%$. Maximální úspěšnosti klasifikace byly v tomto případě stejné $Acc_{max} = 100\%$.

Algoritmy CLONALG a CLONCLAS dosahují stejných nebo lepších výsledků z hlediska maximální úspěšnosti klasifikace Acc_{max} . Pro praktické využití jsou lepší algoritmy se

stabilními výsledky. Algoritmus, který s jedním nastavením parametrů dosáhne jednou vysoké a podruhé nízké úspěšnosti klasifikace je pro praktické využití nevhodný. Z tohoto důvodu lze na základě provedených experimentů doporučit algoritmus Immunos-99, jehož dosažené průměrné výsledky v rámci provedených experimentů jsou ve srovnání s CLONALG a CLONCLAS lepší (zejména při modelování bonity obcí). Pro algoritmus Immunos-99 lze doporučit navrženou variantu dynamického prahování s parametrem, která v případě obou datových souborů dosáhla lepších průměrných výsledků a stejných nebo lepších maximálních hodnot úspěšnosti klasifikace.

Výsledky získané pro použité algoritmy dále ukazují na možnosti jejich optimalizací. Pro algoritmus Immunos-99 lze pro další vývoj doporučit další analýzu chování tohoto algoritmu s cílem zvýšit úspěšnost klasifikace. Motivací je využít toho, že tento algoritmus dosahuje relativně stabilních výsledků. Pro algoritmy CLONALG a CLONCLAS lze v rámci dalšího vývoje doporučit analýzu chování s cílem dosáhnout vyšší stability dosažených výsledků. Pro všechny použité algoritmy lze v rámci dalšího vývoje provést analýzu vlivu nastavení jednotlivých parametrů. Ve srovnání s dosaženými výsledky jiných algoritmů pro použité úlohy lze konstatovat, že algoritmy CLONALG, CLONCLAS a zejména Immunos-99 s dynamickým prahováním s parametrem jsou dobře použitelné pro úlohu klasifikace.

Použitá literatura:

- [1] ASUNCION, A., NEWMAN, D. J. *UCI Machine Learning Repository* [database online]. Irvine, CA: University of California, School of Information and Computer Science, 2007. [citováno 2008-09-09]. Dostupné z <<http://archive.ics.uci.edu/ml/>> <<http://mlr.cs.umass.edu/ml/datasets/>>.
- [2] BRABAZON, A., O'NEILL, M. Credit Classification Using Grammatical Evolution. *Informatica*, 2006, no. 30, pp. 325-335.
- [3] BRENNAN, D., BRABAZON, A. Corporate Bond Rating Using Neural Networks. In *Proceedings of the International Conference on Artificial Intelligence, IC-AI '04, June 21-24, 2004, Las Vegas, Nevada, USA*. CSREA Press, 2004, vol. 1, pp. 161-167.
- [4] BROWLEE, J. *Clonal Selection Algorithms*. Technical Report, no. 070209A, Center for Intelligent Systems and Complex Processes (CISCP), Faculty of Information and Communication Technologies, Swinburne University of Technology, Australia. February 2007.
- [5] BROWLEE, J. *Clonal Selection Theory & Clonalg - The Clonal Selection Classification Algorithm (CSCA)*. Technical Report, no. 2-02, Center for Intelligent Systems and Complex Processes (CISCP), Faculty of Information and Communication Technologies, Swinburne University of Technology, Australia. January 2005.
- [6] BROWLEE, J. *Immunos-81, The Misunderstood Artificial Immune System*. Technical Report, no. 3-01, Center for Intelligent Systems and Complex Processes (CISCP), Faculty of Information and Communication Technologies, Swinburne University of Technology, Australia. January 2005.
- [7] BROWLEE, J. *WEKA Classification Algorithms* [počítačový program]. Ver. 1.6. Sourceforge.NET, 2006 [citováno 2008-09-09]. Dostupné z <<http://sourceforge.net/projects/weka.classalgos/>>.
- [8] CARTER, J. H. The Immune System as a Model for Classification and Pattern Recognition. *Journal of the American Medical Informatics Association*, 2000, vol. 7, no. 1, pp. 28-41.
- [9] CASTRO, L. N., TIMMIS, J. I. *Artificial Immune Systems: A New Computational Intelligence Approach*, London: Springer-Verlag, 2002. 357 p. ISBN 1-85233-594-7.

- [10] CASTRO, L. N., ZUBEN, F. J. Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*. 2002, vol. 6, no. 3, pp. 239-251. ISSN 1089-778X.
- [11] CUTELLO, V. et. al. Clonal Selection Algorithms: A Comparative Case Study using effective Mutation Potentials, In *Proceedings Artificial Immune Systems: 4th International Conference Banff, Alberta, Canada (ICARIS 2005)*. Berlin: Springer Verlag, 2005. pp. 13-28.
- [12] DUCH, W., ADAMCZAK, R., GRABCZEWSKI, K. A New Methodology of Extraction, Optimization and Application of Crisp and Fuzzy Logical Rules. *IEEE Transactions on Neural Networks*, 2001, vol. 12, pp. 277-306. ISSN 1045-9227.
- [13] EGGERMONT, J., KOK, J. N., KOSTERS, W. A. Genetic programming for data classification: Partitioning the search space. In *Proceedings of the 2004 ACM symposium on Applied computing*. New York, NY, USA: Association for Computing Machinery ACM, 2004. pp. 1001-1005. ISBN 1-58113-812-1.
- [14] GARRETT, S. M. How Do We Evaluate Artificial Immune Systems? *Evolutionary Computation*, Summer 2005, vol. 13, no. 2, pp. 145–177. ISSN 1063-6560.
- [15] HÁJEK, P., OLEJ, V. Hierarchical Structure of Fuzzy Inference Systems Design for Municipal Creditworthiness Modeling. *WSEAS Transaction on Systems and Control*, February 2007, vol. 2, no. 2, pp.162-169. ISSN 1991-8763.
- [16] HÁJEK, P., OLEJ, V. Modelling Municipal Rating by Cluster Analysis and Neural Networks. In *Proceedings of the 7-th WSEAS International Conference on Neural Networks, NN 2006, Cavtat, Croatia, June 12-14*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2006, pp.73-78. ISBN 960-8457-46-7.
- [17] HÁJEK, P., OLEJ, V. Municipal Creditworthiness Modelling by Kohonen’s Self-Organizing Feature Maps and LVQ Neural Networks. In *Proceedings of the 9th Eighth International Conference on Artificial Intelligence and Soft Computing, ICAISC 08, Lecture Notes in Artificial Inteligence, Zakopane, Poland, June 22-26*, Springer Berlin Heidelberg New York, 2008, pp. 52-61. ISSN 0302-9743, ISBN 3-540-69572-9.
- [18] HÁJEK, P., OLEJ, V. Municipal Creditworthiness Modelling by Clustering Methods. In *Proceedings of the 10th International Conference on Engineering Applications of Neural Networks, EANN 2007, Margaritis, Illiadis, Eds., Thessaloniki, Greece, August 29-31*. 2007, pp.168-177, ISBN 978-960-287-093-8.
- [19] IGAWA, K., OHASHI, H. Discrimination-based Artificial Immune System: Modeling the Learning Mechanism of Self and Non-self Discrimination for Classification. *Journal of Computer Science*, 2007, vol. 3, no. 4, pp. 204-211. ISSN 1549-3636.
- [20] KOTSIANTIS, S. B. PINTELAS, P.E. Logitboost of Simple Bayesian Classifier. *Informatica*. 2005, vol. 29, no. 4, pp. 53-59. ISSN 0350-5596.
- [21] KVASNIČKA, V. a kol. *Úvod do teorie Neuronových sítí*. Bratislava: IRIS, 1997. ISBN 80-88778-30-1.
- [22] LIU, X., LIU, W. Credit Rating Analysis with AFS Fuzzy Logic. *Advances in Natural Computation, Lecture Notes in Computer Science, Springer-Verlag*, 2005, vol. 3612, pp. 1198-1204. ISSN 0302-9743.
- [23] MAGNI, C.A. Rating and Ranking Firms with Fuzzy Expert Systems: The Case of Camuzzi. *MPPA Paper University Library of Munich, Germany*, 2007, vol. 5889.
- [24] OLEJ, V., HÁJEK, P. Modeling Municipal Rating by Unsupervised Methods. *WSEAS Transactions on Systems*, July 2006, vol. 5, no. 7, pp.1679-1686, ISSN 1109-2777.
- [25] ROMANIUK, S.G. *Fuzzy Rule Extraction for Determining Creditworthiness of Credit Applicants*. Technical report, No. TR20/92, National University of Singapore, 1992.

- [26] SHOLOM, W. M., IOANNIS, K. An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. New York, NY, USA: Morgan Kaufmann, 1989. pp. 781-787. ISBN 1558600949.
- [27] WATKINS, A. *AIRS: A Resource Limited Artificial Immune Classifier*. Mississippi State University, MS. USA, 2001. Master's thesis.
- [28] WATKINS, A. *Exploiting Immunological Metaphors in the Development of Serial, Parallel, and Distributed Learning Algorithms*, University of Kent, Canterbury, UK, 2005. PhD dissertation.
- [29] WATKINS, A., BOGGESS, L. A New Classifier Based On Resource Limited Artificial Immune Systems. In *Proceedings of Congress on Evolutionary Computation, Part of the 2002 IEEE World Congress on Computational Intelligence held in Honolulu, HI, USA, May 12-17*. IEEE Computer Society Press, 2002. pp. 1546-1551.
- [30] WATKINS, A., TIMMIS, J., BOOGESS, L. Artificial Immune Recognition System (AIRS): An Immune Inspired Supervised Machine Learning Algorithm. *Genetic Programming and Evolvable Machines*, 2004, vol. 5, no. 3, pp. 291-317. ISSN 1389-2576.
- [31] WHITE, J., GARRETT, S. M. Improved Pattern Recognition with Artificial Clonal Selection? In *Proceedings Artificial Immune Systems: Second International Conference, September 1-3, 2003, Edinburgh, UK, (ICARIS-2003)*. Berlin: Springer Verlag, 2003. pp.181-193.
- [32] WITTEN, I. H., FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques*. Second Edition. Morgan Kaufmann, 2005. ISBN 0-12-088407-0.
- [33] ZHOU, X. Y., ZHANG, D. F., JIANG, Y. A New Credit Scoring Method Based on Rough Sets and Decision Tree. *Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, Springer-Verlag*, 2008, vol. 5012, pp. 1081-1089. ISSN 0302-9743.

Kontaktní adresa:

Ing. Luděk Kopáček, prof. Ing. Vladimír Olej, CSc.
 USII/FES Univerzita Pardubice
 Studentská 84
 532 10 Pardubice
 E-mail: ludek.kopacek@upce.cz
vladimir.olej@upce.cz