

УДК 004.93

**С. А. Зайцев, С. А. Субботин**

Запорожский национальный технический университет  
ул. Жуковского, 64, 69063 Запорожье, Украина

## **Построение диагностических моделей для бинарных данных на основе отрицательного отбора**

*Проведен анализ методов отрицательного отбора на основе искусственных иммунных систем, пригодных для построения диагностических моделей, работающих с бинарными данными. Проанализированы бинарные метрики, используемые в отрицательном отборе. Предложена модификация метода отрицательного отбора с цензурированием, позволяющая повысить скорость генерации набора детекторов и обеспечивающая при этом высокую точность диагностирования.*

**Ключевые слова:** искусственная иммунная система, отрицательный отбор, бинарные метрики, детектор, диагностическая модель.

### **Введение**

Диагностирование объектов и систем представляет собой одно из основных средств обеспечения и поддержания надежности их функционирования. Для осуществления неразрушающего диагностирования необходимо построить модель зависимости выходного значения (номера класса) от входного набора значений диагностических признаков по набору экспериментальных наблюдений, что позволит в дальнейшем использовать данную модель для автоматизации принятия диагностических решений.

Среди различных технологий, позволяющих автоматизировать процесс диагностирования, особого внимания заслуживают искусственные иммунные системы (ИИС) [1], поскольку они способны самообучаться и обобщать накопленные знания. Диагностические модели, построенные на основе иммунокомпьютинга, позволяют выполнять двухклассовую классификацию диагностируемых объектов по признакам.

В большинстве практических задач применяют вещественное представление признаков. В то же время, в ряде задач диагностирования наблюдения могут быть сведены к бинарному представлению признаков. Для обработки бинарных диагностических признаков целесообразно использовать принципы отрицательного отбора в ИИС, благодаря таким его свойствам, как: эффективность работы с данными, представленными в бинарном виде; высокая точность работы, даже при

© С. А. Зайцев, С. А. Субботин

обучении на небольшом объеме данных; возможность обработки дискретных данных большой размерности. Однако известные модели отрицательного отбора [2] генерируют избыточное количество кандидатов в детекторы, что отрицательно влияет на скорость работы таких моделей. Это обуславливает необходимость разработки модели отрицательного отбора, лишенной описанных выше недостатков.

*Цель работы* заключается в построении модели ИИС на основе принципов отрицательного отбора, которая позволит синтезировать диагностические модели, работающие с бинарным представлением признаков.

## Постановка задачи

Пусть задана выборка  $X \subset U$ , где  $U$  — область допустимых значений входных данных, содержащая  $n$  экземпляров. Каждый  $s$ -й экземпляр  $x^s \in X$ ,  $x^s = \{x_j^s\}$ , характеризуется  $l$  бинарными признаками  $x_j^s$ , где  $s = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, l$ . Каждому экземпляру обучающей выборки  $x^s$  сопоставлен номер класса  $y^s$ . Множество  $y = \{y^s\}$  будет задавать выходной признак. Тогда задача автоматизации построения диагностической модели может быть представлена как задача аппроксимации качественной зависимости номера класса  $y$  от значений признаков экземпляра  $x \in X$  по обучающей выборке данных  $\langle x, y \rangle$ . Применительно к ИИС, необходимо синтезировать модель отрицательного отбора, представленную кортежем  $\langle U, D, match \rangle$ , где  $D$  — набор детекторов,  $match$  — правило, позволяющее классифицировать экземпляры, полагаясь на знания, описываемые набором детекторов.

## Отрицательный отбор

В основе работы модели отрицательного отбора [3] лежит поведение Т-клеток, обеспечивающее терпимость иммунной системы организма к собственным клеткам. При этом Т-клетки обладают способностью распознавать практически любые (неизвестные им ранее) антигены (чужеродные объекты).

Основная задача, решаемая с помощью модели отрицательного отбора, состоит в обнаружении отличий между двумя классами объектов и в проведении дальнейшей двухклассовой классификации [4].

С точки зрения ИИС важной частью модели отрицательного отбора является набор детекторов (часто называемый также множеством детекторов). Детекторы представляют собой объекты, способные классифицировать неизвестные экземпляры, как «свои» (self), так и «чужие» (non-self). Классификация происходит на основе правила сопоставления (метрики). В случае, когда значение результата правила сопоставления истинно, то говорят, что детектор активизирован, и сопоставляемый с ним объект следует отнести к классу «чужих». В противном случае сопоставляемый объект считают «своим».

Вначале требуется определить множество «своих» экземпляров  $S$  в пространстве признаков  $U$ . Затем на основе экземпляров множества  $S$  формируется набор детекторов  $D$ . Набор создается таким образом, чтобы каждый детектор,

принадлежащий множеству  $D$ , не активизировался при сопоставлении с любым из экземпляров множества  $S$ .

После того, как набор  $D$  сформирован, модель считается обученной. С этого момента ей на вход можно подавать экземпляры, которые необходимо диагностировать, после чего модель, используя набор детекторов  $D$ , принимает решение об отнесении экземпляра с неизвестной классификацией к классу «своих» или «чужих».

При использовании модели отрицательного отбора на практике важно определить, детекторы какого рода будут использоваться для описания пространства признаков, и какие метрики будут к ним применяться для определения класса объекта.

### Метрики для работы с бинарными детекторами

Для сравнения двух бинарных детекторов применялись различного рода метрики, позволяющие определить степень подобия двух битовых строк [5].

**Правило  $r$ -последовательных битов** ( $r$ -contiguous rule, RCB-rule) [6] использовалось изначально в модели отрицательного отбора. Метод генерации детекторов для модели отрицательного отбора оперировал строками фиксированной длины. Так, для двух строк, представленных в виде последовательности из  $n$  битов  $x = \{x_1, x_2, \dots, x_n\}$  и  $d = \{d_1, d_2, \dots, d_n\}$  правило выглядит следующим образом:

$$match(d, x) = \begin{cases} 1, & \exists i, i \leq n-r+1, \forall j, i \leq j \leq i+r-1 : x_j = d_j, \\ 0, & \text{в противном случае.} \end{cases}$$

Иными словами, две строки совпадают, если существует такое окно размером  $r$ , в пределах которого все биты обеих строк совпадают.

Данная метрика отличается своей простотой и используется в оригинальном методе генерации детекторов для модели отрицательного отбора, получившем свое дальнейшее развитие в линейном и «жадном» методах генерации детекторов. Все эти методы ограничиваются использованием бинарной формы представления детекторов и RCB-правила.

**Метрика R-chunks** [7] является более общей, по сравнению с RCB-метрикой, т.е. любой детектор, оперирующий RCB-правилом, может быть представлен в виде множества  $r$ -chunks детекторов. Для двух строк  $x = \{x_1, x_2, \dots, x_n\}$  и  $d = \{d_1, d_2, \dots, d_m\}$  длиной  $n$  и  $m$  соответственно,  $n \leq m$ , правило  $r$ -chunks можно представить как

$$match(d, x) = \begin{cases} 1, & \forall j, i \leq j \leq i+m-1 : x_j = d_j, \\ 0, & \text{в противном случае,} \end{cases}$$

где  $i$  определяет позицию начала отрезка строки (chunk).

Правило  $r$ -chunks позволяет повысить точность работы метода отрицательного отбора.

**Метрика Хемминга** применялась в [8] в качестве правила сопоставления

$$match(d, x) = \begin{cases} 1, & \sum_{i=1}^n |x_i \oplus d_i| \geq r, \\ 0, & \text{в противном случае,} \end{cases}$$

где  $n$  — длина битовой строки;  $\oplus$  — операция «исключающее ИЛИ»;  $r$  — порог срабатывания правила,  $0 \leq r \leq n$ .

**Метрика R&T (Rogers&Tamiro)**, как отмечено в [9], отличается более высокой точностью, по сравнению с метрикой Хемминга. R&T-правило для двух битовых строк одинаковой длины  $n$   $x = \{x_1, x_2, \dots, x_n\}$  и  $d = \{d_1, d_2, \dots, d_n\}$  определяется как

$$match(d, x) = \begin{cases} 1, & \frac{\sum_{i=1}^n x_i \oplus d_i}{\sum_{i=1}^n x_i \oplus d_i + 2 \sum_{i=1}^n x_i \oplus d_i} \geq r, \\ 0, & \text{в противном случае.} \end{cases}$$

Однако на сегодняшний день для рассмотренных метрик не существует более оптимального метода генерации детекторов, чем случайная генерация и цензурирование.

**Метрика Левенштейна** [8] может считаться обобщением метрики Хемминга. Ее значение определяется минимальным количеством изменений, необходимых для преобразования одной бинарной строки в другую. При этом изменения могут быть следующего вида: вставка символа, удаление символа, замена одного символа (бита) другим из алфавита. В некоторых вариациях метрика Левенштейна считает замену нескольких смежных символов одной операцией.

## Метод отрицательного отбора с цензурированием

Рассмотрим модель [3], реализующую парадигму отрицательного отбора. Работа данной модели осуществляется в два этапа.

1. Генерация набора детекторов. Для этого случайно сгенерированные кандидаты  $C$ , представленные в виде битовых строк, проходят через этап цензурирования и те, которые не отсеиваются на этом этапе, попадают в набор детекторов  $D$ .

2. Мониторинг. На этом этапе данные, соответствующие «своим» экземплярам, сравниваются с детекторами, используя одну из метрик (как правило, РСВ-метрику). Если хотя бы один из детекторов при этом активизируется, т.е.  $\exists d \in D, s \in S : match(d, s) = 1$ , считается, что произошли изменения во множестве  $S$ .

На практике оказывается [4], что множество  $D$  относительно небольшой мощности способно обеспечить достаточно высокую вероятность обнаружения изменений во входных данных. Более того, при неизменном количестве детекто-

ров объем «своих» экземпляров может увеличиваться без снижения точности классификации.

Метод генерации детекторов представляется чрезвычайно ресурсоемким, а потому очень важно своевременно осуществить останов во избежание генерации избыточного количества детекторов. Также оценка вероятности активации детектора в наборе позволяет управлять параметрами метода.

Вероятность того, что два строковых детектора совпадут на  $r$  смежных битах, определяется как

$$P_m \approx m^{-r} \frac{(l-r)(m-1)}{m+1}, \quad (1)$$

где  $m$  — количество символов в алфавите (например, для бинарных строк  $m = 2$ );  $l$  — длина каждой из строк;  $r$  — количество совпадающих смежных символов. Данная аппроксимация рекомендуется [3] для случаев, когда  $m^{-r} \ll 1$ .

Отметим, что вероятность совпадения двух строк  $P_m$  снижается линейно в зависимости от  $l$  и экспоненциально — в зависимости от  $r$ . Еще более резкое снижение  $P_m$  наблюдается при увеличении размерности алфавита. Так, для бинарного алфавита  $\{0, 1\}$  вероятность совпадения 8 бит на 32-битных строках составляет около 5 %, однако для 128-символьного алфавита (например, ASCII символов) совпадение 8 символов из 32 составляет около  $3 \cdot 10^{-16}$ .

В [3] установлено, что для обнаружения случайных изменений в определенном количестве  $N_S$  «своих» экземпляров с заданной вероятностью  $P_f$  при вероятности совпадения двух строк, равной  $P_m$ , требуется сгенерировать около  $N_C$  случайных детекторов, где

$$N_C = \frac{-\ln(P_f)}{P_m (1 - P_m)^{N_S}}. \quad (2)$$

Также показано, что минимальное количество кандидатов в детекторы  $N_C$ , требуемых для обеспечения заданной точности классификации, может быть получено при  $P_m \approx \frac{1}{N_S}$ .

Таким образом, предложенный в [3] метод является настраиваемым, поскольку позволяет оценить количество требуемых детекторов для заданной точности классификации, и, с другой стороны, установить достаточную точность классификации, исходя из максимально возможного числа детекторов (например, ограниченного объемом оперативной памяти вычислительного узла). Также метод хорошо работает в распределенной среде, поскольку вероятность того, что в системе из  $N_S$  вычислительных узлов встретится «чужой» экземпляр определяется как  $P_f^* = P_f^{N_S}$ .

Вычислительная сложность метода при последовательной организации вычислений оценивается [3] таким образом:

$$O\left(\frac{-\ln(P_f)}{P_m(1-P_m)^{N_S}}\right)N_S.$$

Пространственная сложность метода в [3] оценивается как  $O(l \cdot N_S)$ .

Существуют методы, также реализующие парадигму отрицательного отбора, которые позволяют снизить вычислительную сложность [10], однако их использование затруднено высокой пространственной сложностью.

Основным недостатком данного метода является слишком большое количество генерируемых на стадии обучения модели кандидатов в детекторы, в результате чего модель довольно медленно адаптируется к поступающим входным данным. Решение этой проблемы реализовано в предложенном модифицированном методе отрицательного отбора.

### Модифицированный метод отрицательного отбора

Для устранения недостатков базового метода отрицательного отбора предлагается его модифицировать за счет генерации кандидатов в детекторы, основанной на вероятностном распределении значений диагностических признаков. Предлагаемая модификация метода отрицательного отбора с цензурованием позволяет получить детекторы, соответствующие «чужим» экземплярам на более ранних итерациях. Основная идея предложенного метода заключается в том, что признаки в «своих» и «чужих» экземплярах не всегда принимают значения «1» или «0» с равной вероятностью. Иными словами, если известно распределение значений признаков для «чужих» экземпляров, то сгенерированные согласно этому распределению псевдослучайные детекторы с большей вероятностью будут соответствовать «чужим» элементам.

Пусть  $P_i$  — вероятность того, что  $i$ -й бит в «своем» экземпляре принимает значение «1». Известно, что в пространстве признаков содержится одинаковое количество экземпляров с нулевым  $i$ -м битом и с ненулевым. Следовательно,  $i$ -й бит некоторого «чужого» экземпляра равен «1» с вероятностью  $P_i' = 1 - P_i$ .

Таким образом, можно генерировать псевдослучайные детекторы на основе статистических характеристик только «своих» экземпляров, что соответствует принципам отрицательного отбора.

Пусть задано множество «своих» битовых строк  $S \subset U$ . Тогда для генерации набора детекторов метод должен включать следующие этапы.

1. Подсчитать количество ненулевых бит в «своих» экземплярах. Для каждого экземпляра  $s \in S$ :  $k_i = k_i + 1$ , если  $s_i = 1$ ,  $i = 1..l$ , где  $k$  — количество ненулевых бит в «своих» экземплярах;  $i$  — порядковый номер бита.

2. Оценить вероятности, с которыми биты «своих» экземпляров принимают ненулевые значения:  $P_i = \frac{k_i}{|S|}$ .

3. Оценить вероятности, с которыми биты «чужих» экземпляров принимают ненулевые значения:  $P'_i = 1 - P_i$ .

4. Используя полученные оценки вероятностей, генерировать псевдослучайные детекторы  $d$  :

$$d_i = \begin{cases} 1, & \text{rand} < P'_i, \\ 0, & \text{в противном случае,} \end{cases}$$

где  $\text{rand}$  — функция, возвращающая случайные числа в диапазоне  $[0, 1)$ .

Выполнение данного этапа происходит до тех пор, пока не будет сгенерировано достаточное количество детекторов.

В результате работы метода будет получен набор детекторов  $D$ , который затем подвергается цензурированию (так же, как и в оригинальном методе).

Однако в таком виде модифицированный метод генерирует детекторы с более высокой степенью подобия к «чужим» экземплярам, хотя в последствии его точность остается практически неизменной.

С целью повышения точности работы метода предлагается добавить еще один этап, выполняющий своего рода обратную связь. На этом этапе значения вероятностей пересчитываются после цензурирования в зависимости от того, был ли кандидат добавлен в набор детекторов. Так, если некоторый псевдослучайный кандидат  $d$  был отвергнут (иными словами, был отнесен к классу «своих») и  $d_i = 1$ , установить  $k_i = k_i + 1$ . После этого следует выполнять этапы 2–4 для генерации следующих детекторов.

Таким образом, если для текущего распределения вероятностей значений признаков случайный кандидат в детекторы не удовлетворяет критерию отбора, то вероятности значений признаков изменяются таким образом, чтобы новые псевдослучайные кандидаты имели больше шансов попасть в набор детекторов.

Поскольку предложенный метод не учитывает порядок битов, то наиболее целесообразно использовать метрику Хемминга для вычисления степени подобия двух детекторов.

## Эксперименты и результаты

Для проведения экспериментов было разработано программное обеспечение на языке Go [11]. Оно включает в себя программную реализацию метода отрицательного отбора на основе описанных выше метрик, а также реализацию модифицированного метода. Данное программное обеспечение позволяет строить модели зависимостей выходных значений от набора значений признаков входных бинарных данных с использованием принципов иммунокомпьютинга.

Базовый и модифицированный методы отрицательного отбора исследовались при решении задачи построения диагностической модели зависимости показателя

качества жизни (КЖ) от сочетания лекарственных средств [12].

Понятие качества жизни подразумевает то, насколько человек может чувствовать себя полноценно в обществе, семейной жизни, при выполнении своих профессиональных обязанностей в условиях наличия хронического заболевания. Использование модели зависимости КЖ от сочетания лечебных мероприятий позволяет проводить оценку их эффективности, что дает возможность упростить и удешевить процесс лечения.

Для оценки и сравнения эффективности лекарственных средств использовалась выборка данных [12] о лечении 117 больных хроническим обструктивным бронхитом (ХОБ) в Запорожской областной клинической больнице, для лечения которых применялись в различных сочетаниях 95 лекарственных средств и лечебных процедур. Каждый признак принимает значение «0» или «1» в зависимости от того, использовалось ли соответствующее средство при лечении больного. В качестве выходного признака, характеризовавшего эффективность лечения, использовалась нормированная разность суммарных показателей качества жизни (СПКЖ) пациентов при выписке и поступлении [13, 14].

Поскольку принцип отрицательного отбора применим только для двухклассовой классификации, то целесообразно использовать дихотомию и разбить множество  $X$  на диапазоны по значению  $y$ . В рассматриваемой задаче выборка была разбита на 7 диапазонов, содержащих от 8 до 27 экземпляров, и для каждого диапазона выполнялось построение отдельной модели на основе ИИС. Для простоты в дальнейшем будем рассматривать построение модели отрицательного отбора для одного отдельного диапазона.

При решении задачи выполнялось построение моделей отрицательного отбора с использованием метрик:  $gcb$ ,  $r$ -chunks и Хемминга. Также варьировались значения порога  $r$  для данных метрик.

Согласно (1) и (2) для обучения и тестирования модели на 95-битных экземплярах для получения 95 % точности потребуется сгенерировать более 100 тыс. детекторов, что существенно снизит быстродействие модели, особенно этапа цензурирования. В связи с этим было принято решение снизить размерность задачи за счет исключения неинформативных признаков. В результате была получена тестовая выборка, содержащая экземпляры, описанные битовыми строками размерностью 17 битов. Обучение производилось на 60 % «своих» экземпляров, чужие экземпляры в обучении не участвовали.

На рис. 1 представлено изменение точности классификации и объема множества детекторов в зависимости от используемой метрики и ее порогового значения. Из рис. 1 видно, что объем набора детекторов возрастает, а точность классификации снижается прямо пропорционально пороговому значению метрики  $r$ , а также то, что метрики  $gcb$  и Хемминга дают наиболее точные результаты.

С помощью метода отрицательного отбора можно получить 80-процентную точность классификации при небольшом объеме множества детекторов. Однако данный метод в процессе работы генерирует слишком большое число кандидатов в детекторы, что усложняет цензурирование и отрицательно влияет на время работы метода.

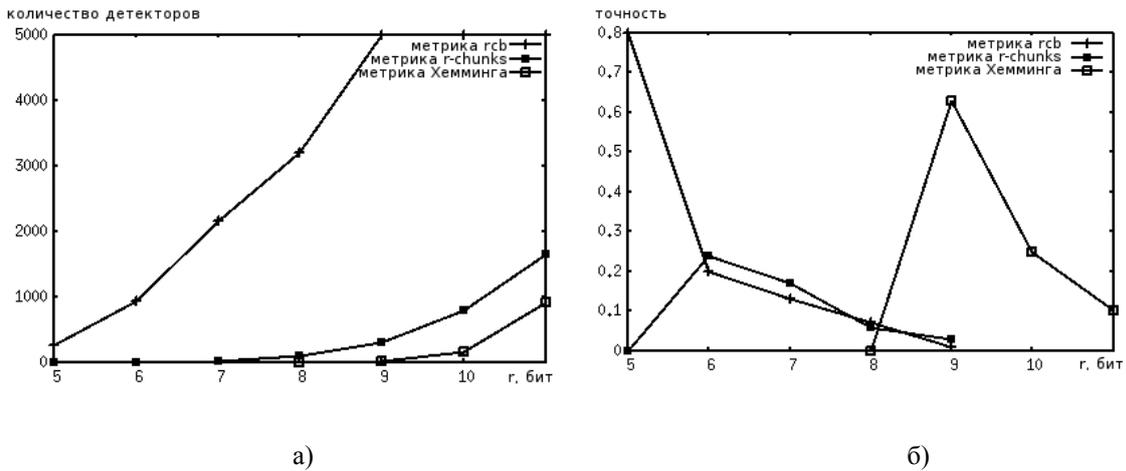


Рис. 1. Влияние вида метрики и ее порогового значения:  
а) на объем множества детекторов; б) на точность классификации

На рис. 2,а представлен график изменения точности классификации предложенного и базового методов отрицательного отбора [4]. Динамика изменения точности работы метода для такой его модификации представлена на рис. 2,б. Модифицированный метод обеспечивает точность классификации до 95 %, при этом он формирует в среднем наборы из 4–5 детекторов и производит около 120 кандидатов в детекторы, благодаря чему работает значительно быстрее базового метода отрицательного отбора.

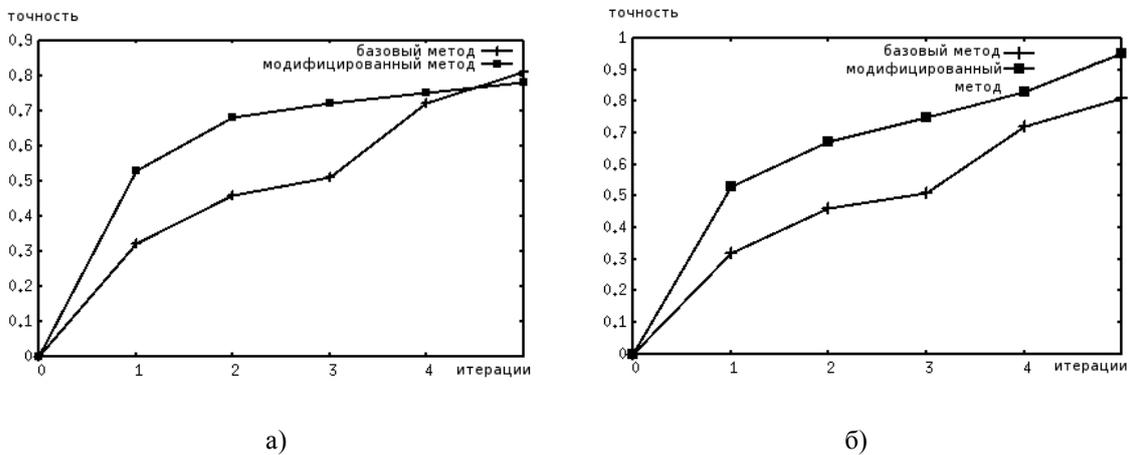


Рис. 2. Изменение точности работы модифицированного метода отрицательного отбора:  
а) без обратной связи; б) с обратной связью

## Заключение

С целью решения актуальной задачи автоматизации построения систем технического диагностирования разработано математическое обеспечение, позво-

ляющее синтезировать диагностические модели на основе парадигмы отрицательного отбора.

*Научная новизна работы* заключается в том, что получил дальнейшее развитие метод отрицательного отбора, модифицированный таким образом, что генерация кандидатов в детекторы осуществляется на основании вероятностного распределения значений признаков экземпляров обучающей выборки. Данная модификация позволяет повысить скорость обучения диагностической модели за счет формирования набора детекторов за меньшее время.

*Практическая ценность работы* состоит в том, что разработано программное обеспечение для построения диагностических моделей для бинарных данных. На основе разработанного программного обеспечения получена модель зависимости значений КЖ от комбинаций назначаемых лекарственных средств, что позволяет прогнозировать изменение СПКЖ, а также определять наиболее эффективное сочетание лекарственных средств и лечебных процедур.

Разработанное математическое обеспечение успешно прошло стадию тестирования, обладает высокими показателями точности и может быть рекомендовано для использования на практике. Дальнейшие исследования могут быть сосредоточены на анализе влияния различных критериев останова метода отрицательного отбора на объем набора детекторов и точность работы построенной модели зависимости.

1. *Dasgupta D.* Artificial Immune Systems and Their Applications / Ed. D. Dasgupta. — New York: Springer-Verlag, 1998. — 306 p.
2. *Ji Z.* Revisiting Negative Selection Algorithm / Zhou Ji, Dipankar Dasgupta // Evolutionary Computation. — 2007. — Vol. 15, N 2 (Summer). — P. 223–251.
3. *Forrest S.* Self Nonself Discrimination in a Computer / S. Forrest, A.S. Perelson, L. Allen, R. Cherukuri // The 1994<sup>th</sup> IEEE Symposium on Research in Security and Privacy: proceedings, 1994, California. — Los Alamitos: IEEE Computer Society Press, 1994. — P. 202–212.
4. *Dasgupta D.* Artificial Immune Systems in Industrial Applications / D. Dasgupta, S. Forrest // The International Conference IPMM'99: Proceedings, 10–14 of July, 1999, Honolulu, Hawaii, USA. — New York, USA: Wiley Computer Publishing, 1999. — P. 121–135.
5. *Gonzalez F.* The Effect of Binary Matching Rules in Negative Selection / F. Gonzalez, D. Dasgupta, J. Gomez // Genetic and Evolutionary Computation Conference: proceedings, 9–11 of July, 2003, Chicago (USA). — Berlin-Heidelberg: Springer-Verlag, 2003. — P. 195–206.
6. *D'haeseleer P.* An Immunological Approach to Change Detection: Algorithms, Analysis, and Implications / P. D'haeseleer, S. Forrest, P. Helman // The 1996<sup>th</sup> IEEE Symposium on Computer Security and Privacy: proceedings, 6–8 of May, 1996, Oakland. — Los Alamitos: IEEE Computer Society Press, 1996. — P. 110–119.
7. *Balthrop J.* Coverage and Generalization in an Artificial Immune System / J. Balthrop, F. Esponda, S. Forrest, M. Glickman // Genetic and Evolutionary Computation Conference: proceedings, 9–13 of July, 2002, New York, USA. — San Fransisco, USA: Morgan Kaufmann, 2002. — P. 3–10.
8. *Farmer J.* The Immune System, Adaptation, and Machine Learning / J. Farmer, N. Packard, A. Perelson // Physica D: Nonlinear Phenomena. — 1986. — Vol. 2. — P. 187–204.

9. Harmer P.K. An Artificial Immune System Architecture for Computer Security Applications / P.K. Harmer, P.D. Williams, G.H. Gunsch, G.B. Lamont // IEEE Transactions on Evolutionary Computation. — June, 2002. — Vol. 6, N 3. — P. 252–280.

10. Wierzchon S.T. Generating Optimal Repertoire of Antibody Strings in an Artificial Immune System / S.T. Wierzcho // Symposium on Intelligent Information Systems IIS'2000: proceedings, 12–16 of June, 2000, Bystra, Poland. — Berlin-Heidelberg: Physica-Verlag, 2000. — P. 119–133.

11. Pike R. The Go Programming Language. Effective Go [Electronic resource]: specification / Rob Pike, Ken Thompson. — Reference: [http://golang.org/doc/effective\\_go.html](http://golang.org/doc/effective_go.html).

12. Евченко Л.Н. Анализ эффективности методов лечения на основе эвристического и нейросетевого подходов / Л.Н. Евченко, В.И. Кривенко, С.А. Субботин // Моделирование неравновесных систем – 2002: материалы семинара, октябрь 2002, Красноярск. — Красноярск: ИПЦ КГТУ, 2002. — С. 60–61.

13. Кривенко В.И. Нейросетевое моделирование суммарного показателя качества жизни больных хроническим обструктивным бронхитом в ассоциации с клиническими особенностями течения заболевания / В.И. Кривенко, Л.Н. Евченко, С.А. Субботин // Вестник новых медицинских технологий. — 2001. — Т. VIII, № 4. — С. 7–10.

14. Кривенко В.И. Нейросетевое моделирование показателя качества жизни для диспансерного учета пациентов / В.И. Кривенко, Л.Н. Евченко, С.А. Субботин // Моделирование неравновесных систем – 2001: материалы семинара, октябрь 2001, Красноярск. — Красноярск: ИВМ СО РАН, 2001. — С. 79–80.

Поступила в редакцию 15.12.2010