Sistema Inmune Artificial para la Detección de Comportamientos Anómalos de Usuarios en Sistemas Informáticos

César Byron Guevara Maldonado, Matilde Santos P, Victoria López
cesargue@ucm.es, msantos@ucm.es, vlopez@fdi.ucm.es

Facultad de Informática, Universidad Complutense de Madrid

Abstract. Esta este trabajo presenta el desarrollo y aplicación inicial de un algoritmo la dinámico de detección de anomalías en sistemas de informacón gubernamental. Para el desarrollo del algoritmo due necesario utilizar información del comportamiento de varios usuarios que ejecutan multitud de tareas dentro del sistema durante un tiempo determinado. Las principales aportaciones de este trabajo son: el proceso de desarrollo de un modelo de datos dinámico, clasificación de tareas más ejecutadas y las poco ejecutadas por cada usuario, además, la aplicación de un algoritmo de sistemas inmunes artificiales como la selección negativa para la generación de secuencias detectores binarios de tareas anómalas para la posterior detección de comportamientos peligrosos de los usuarios aplicando el algoritmo de búsqueda local Knuth Morris Pratt (KMP).

1 Introducción

El sistema inmune natural ha tenido gran éxito en la protección de los seres vivos contra una amplia variedad de patógenos como lo presenta Tizard en [1]. El Sistema Inmune Artificial (AIS) siempre ha sido una inspiración para el desarrollo de modelos computacionales para resolver diversos problemas incluyendo el diagnóstico de fallas, detección de virus y detección de fraudes hipotecarios como lo presenta Dasgupta en[2]. Específicamente, el sistema inmunológico utiliza dos principios fundamentales que se incluyen en la teoría de redes inmunes que son los mecanismos de selección negativa y los principios de selección clonal. En todos estos ámbitos, la detección de anomalías es un área de investigación importante en la aplicación de un sistema inmune artificial (AIS) que ha sido probado con excelentes resultados. El objetivo más importante de la detección de anomalías es detectar comportamientos no frecuentes, mal uso y abuso de los sistemas informáticos por parte de los usuarios del sistema o intrusos internos o externos. En la actualidad existen muchos sistemas de detección de intrusiones basados en red (IDS) y en host (HIDS) que se han desarrollado utilizando diversos enfoques.

Este trabajo se centra en el estudio de la detección de anomalías en un sistema informático aplicando un conjunto de detectores anómalos a los datos de comportamiento de los usuarios. De ahí que la detección de anomalías de los

datos en secuencias de ejecución de tareas, ya que es un tema importante de investigación. Existen muchos trabajos sobre esta área de investigación de las técnicas de detección de anomalías como Lazarevic en [3] y Chandola en [4] que buscan objetos anormales que son diferentes de los objetos normales.

Estamos interesados en la detección de anomalías en secuencias discretas para encontrar posibles intrusiones, fraudes, fallos o la fuga de datos. La detección de anomalías para las secuencias discretas no es una tarea fácil, ya que implica el análisis de la secuencia normal de datos para detectar posibles anomalías. El principal problema para detectar anomalías en secuencias de tareas es la gran cantidad de datos para el procesamiento y el desarrollo del modelo dinámico adaptable a la conducta humana.

El documento está organizado de la siguiente manera: la sección 2 describe brevemente el AIS para la detección de anomalías, detección de fuga de datos y detección de intrusos que son las obras más importantes en esta área que han sido presentadas por varios autores. La sección 3, muestra los métodos y materiales utilizados como el algoritmo de selección negativa, secuencia algoritmo de búsqueda, además, la forma de aplicación para la detección de anomalías. En la sección 4, se detallan los objetivos de la investigación y las aportaciones al mismo. Sección 5 describe del experimento realizado en este trabajo y un análisis de los resultados. Por último, las conclusiones se han extraído de este artículo y trabajos futuros.

2 Trabajos relacionados

Varios trabajos han sido desarrollados para detectar anomalías en la operación sobre información de llamadas al sistema tal como se presenta en Forrest en[5] y Gao en [6].

En el trabajo presentado por Helman en [7] propone un ranking de cada secuencia comparando la frecuencia que se conoce va a ocurrir las trazas normales y la frecuencia con la que se espera que ocurran las intrusiones. En el artículo de Javitz en [8] utiliza las distribuciones estadísticas para definir el comportamiento normal y anormal.

Por otra parte, los trabajos dirigidos a la detección de anomalías presentado por Mykerjee [9] describe los muchos perfiles de las actividades normales de los usuarios, los sistemas, los recursos del sistema, tráfico de red, servicios y detecta intrusiones mediante la identificación de desviaciones significativas de los patrones de comportamiento normales observados a partir de perfiles.

3 Métodos y Materiales

En esta sección se presentan los principales algoritmos aplicados en nuestro trabajo. Los algoritmos utilizados son:

- Algoritmo de selección negativa (NSA).
- Algoritmo Knuth Morris Pratt (KMP).

3.1 Sistema Inmune Artificial con Selección Negativa

El AIS se basa en el funcionamiento del sistema inmune humano, que es capaz de reconocer en una forma muy eficiente cualquier agente patógeno. Esta es una teoría inmune clásica para entender el sistema inmunológico como un sistema que identifica a self (agente propio) o a non self(agente patógeno)[12].

3.2 Algoritmo de Knuth Morris Pratt

El objetivo principal de este algoritmo es encontrar una cadena dentro otra cadena. En un patrón P para cada posición i, $spi(p)$ se dice que es la longitud del sufijo más largo de $P[1, 2i]$, que coincide con el prefijo P. Es similar a navegar dentro de la cadena y que realiza sus comparaciones de izquierda a derecha. También calcula los desplazamientos máximos posibles de izquierda a derecha para el patrón P, como lo presenta Gawrychowski en [11].

4 Algoritmo de Detección de Anomalías

4.1 Datos del estudio y estructura de datos

Los datos seleccionados para este trabajo fueron recogidos de un sistema de gobierno de la República de Ecuador. Esta información es confidencial y estos datos se han codificado para salvaguardar la integridad de su información. El conjunto de datos utilizados en este trabajo se ha generado mediante la captura de las tareas ejecutadas por los usuarios. Este conjunto de datos consta de diez conjuntos diferentes de información sobre la ejecución de las tareas en sesiones de cada usuario. Las sesiones grabadas durante el período de 2011 a 2013 son 15.571 y el conjunto de sesiones de pruebas con sesiones anómalas son 5312.

El problema principal es muy difícil distinguir entre comportamiento normal y comportamiento anormal porque las tareas ejecutadas por los usuarios depende de la carga de trabajo o asignaciones de cada usuario. Este sistema tiene siete tablas principales en la base de datos ($T1, T2..., T7$), en esta base de datos es posible ejecutar las cuatro operaciones sql que son: Insertar(1), Modificar (2), Eliminar (3) y Buscar(4). Las tareas están codificadas en 28 tareas genéricas más el inicio de sesión. Cada tarea tiene un código "Tsk" más un número para identificar cada tarea.

El comportamiento del usuario es dinámico y diferente de otro usuario, por esta razón tenemos que crear una estructura de datos que describa el comportamiento humano dentro de un sistema informático. Esta estructura está formada por un inicio de sesión (Tsk0) y una o más tareas ejecutadas en forma secuencial. Esta estructura no tiene tamaño fijo como se muestra en la figura 1.

El conjunto de tareas se define como $Tsk = \{Tsk_1, Tsk_2, Tsk_3, ..., Tsk_n\}$ donde n es el $n$úmero de la tarea. El conjunto de posibles tareas ejecutadas por el usuario U se define $Tsk^U = \{Tsk_1, Tsk_2, Tsk_3, ..., Tsk_m\}$, donde m es el número de tareas ejecutadas por un usuario específico. Una sesión de usuario S contiene una o varias tareas ejecutadas por el usuario y se define como $S^U =$

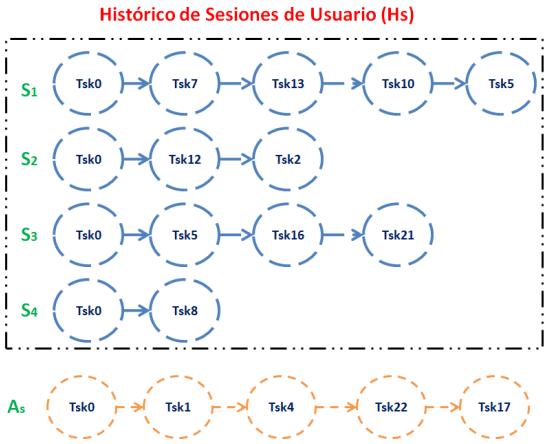

Fig. 1. Estructura de datos del comportamiento de usuario en un sistema informático.

$\{Tsk_1^U, Tsk_2^U, Tsk_3^U, ..., Tsk_m^U\}$. El conjunto de posibles sesiones se define como $Hs^U = \{S_1^U, S_2^U, S_3^U, ..., S_d^U\}$, donde d es el número de sesiones realizadas por el usuario, estas sesiones se llama "Sesiones pasivas" donde $Hs^U \subset S^U$. Una nueva sesión de usuario se define como Sesión Activa As^U, la cual puede contener tareas Tsk^U como tareas que no ha sido ejecutadas por el usuario U.

4.2 Objetivo del Experimento

La detección de anomalías AIS que se propone describe las siguientes contribuciones:

- Detectar comportamientos anómalos de los usuarios dentro de los sistemas informáticos.
- Creación de una estructura dinámica de tareas del comportamiento de los usuarios.
- Identificar de las tareas más ejecutadas por el usuario para detectar el comportamiento anormal.
- Generar secuencias de tareas anómalas utilizando el algoritmo de selección negativa como en lo presenta Kim, para detectar comportamientos anómalos utilizando la formulación de Forrest.
- Aplicar el algoritmo KMP para detectar el comportamiento anómalo de los usuarios.

En la siguiente sección se describe el proceso del algoritmo propuesto para aplicar la selección negativa con la nueva estructura de datos.

4.3 Proceso del algoritmo propuesto

La detección secuencial de tareas anormales podría resumir en los siguientes tres pasos:

Paso 1. Ranking de usuarios tareas más ejecutadas En esta etapa el objetivo es agrupar las tareas ejecutadas (Tsk) en 3 subgrupos, estos son: tareas más ejecutadas (ME), tareas ejecutadas por debajo de la media (MDE) y las tareas no ejecutadas (NE). Para hacer esta agrupación es necesario utilizar el conjunto de datos Hs^U, debido a que cada usuario tiene grupos específicos de tareas Tsk^U acuerdo a su comportamiento. El grupo ME establece las tareas más populares ejecutados por el usuario. Estas tareas son buena parte del conjunto Hs^U.

El segundo grupo MDE son las tareas ejecutadas con poca frecuencia por el usuario, este grupo se identifica como tareas complementarias y no proporciona mucha información del comportamiento del usuario.

El último grupo NE son las tareas no ejecutadas por el usuario y estas tareas son probablemente las tareas anómalas.

Este procedimiento realiza un recuento de las tareas ejecutadas Tsk^U de las sesiones S^U, definido como $XTask_n$, donde n es el número de la tarea. Esos valores se deben ordenar en forma descendente. Después, el número total de ejecuciones de cada una de las tareas $Task_n$ se utiliza para calcular la media de los valores definidos como Avg_{Med}. Si $Xtask_n > Avg_{Med}$ entonces esta tarea $Task_n$ será etiquetada como ME, de lo contrario esta tarea será etiquetada como MDE. Si $Xtask_n$ es igual a cero, es decir que no contiene una sola ejecución de $Xtask_n$ se etiqueta como NE. Finalmente, la tabla obtenida se almacena en la memoria para llevar a cabo nuevas actividades en los siguientes pasos.

Paso 2. Proporción de las tareas en las sesiones de usuario En el segundo paso, el procedimiento es determinar el porcentaje de tareas ME, MDE y NE en las sesiones S^U del conjunto de datos Hs^U, como también de la sesión a ser evaluada As^U.

Este conjunto de valores corresponde al porcentaje de cada tarea de usuario ejecutado en el conjunto de datos Hs^U definidos como $Pr(Me^b_m)$ para las tareas más ejecutadas ME. Y $Pr(Med^b_m)$, para las tareas MDE, donde b es el número de la tarea y m es el número de la sesión a ser evaluada. El cálculo del valor de porcentaje de tareas ME y tareas MDE para una sesión S^U o As^U se aplica las siguientes eciaciones:

$$Pr(Me^b_m) = \frac{N_{Me}}{N_{tasks}} 100\%$$

N_{Me} es el número de tareas ME dentro de una sesión (S^U o As^U).

$$Pr(Med^b_m) = \frac{N_{Med}}{N_{tasks}} 100\%$$

N_{Med} es el número de tareas MED dentro de una sesión (S^U o As^U).

N_{tasks}es el número total de tareas dentro de una sesión (S^U o As^U).

Posteriormente el procedimiento calcula un valor máximo y mínimo de $Pr(Me^b_m)$, definido como $Rmax_{Me}$ y $Rmin_{Me}$. También el valor máximo y mínimo de $Pr(Med^b_m)$ que está definido como $Rmax_{Med}$ y $Rmin_{Med}$.

Para determinar el porcentaje de tareas dentro As^U es necesario aplicar la ecuaciones 1 y 2 obteniento los porcentajes $Pr(Med^b)^{As^U}$ y $Pr(Me^b)^{As^U}$, en el

caso de que existiese tareas NE se determina un umbral $Pr(Ne^b)^{As^U}$ mayor a cero y menor a 1 para identificar su presencia en la siguiente parte del algoritmo. Con estos umbrales en los siguientes pasos podemos evaluar una o más sesiones As^U para conocer si están dentro de los parámetros de un comportamiento normal de usuario.

Paso 3.Generación de secuencias anormales La generación de secuencias anomalas propuesta consta de 3 fases principales:

- Fase 1: Generar aleatoriamente la población inicial B_0. Esta fase se ilustra en la figura 2a.
- Fase 2: Definición de la porción de espacio normal (itself). Esta fase se ilustra en la figura 2b.
- Fase 3: La eliminación de todos los detectores que se solapan en la región normal (itself). Esta fase se ilustra en la figura 2c y obtención de detectores anómalos Dan, como lo presenta la figura 2d.

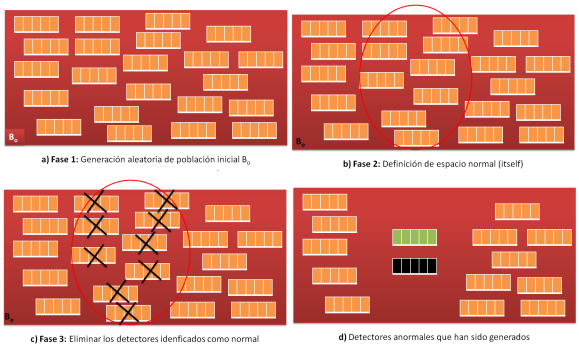

Fig. 2. Fases del sistema de detección de anomalías.

Fase 1. Generación aleatoria de población inicial B_0

En esta fase se presenta el uso de un sistema inmunológico artificial mediante la aplicación de selección negativa para generar secuencias de tareas anómalas de conjunto de datos históricos Hs^U con las proporciones obtenidas en el apartado anterior que son $Pr(Me^b)^{As^U}$, $Pr(Med^b)^{As^U}$ y $Pr(Ne^b)^{As^U}$. Para generar la población inicial B_0 se debe cuenta las siguientes condiciones:

Si $Rmax_{Me} > Pr(Me^b)^{As^U} \geq Rmin_{Me}$, $Rmax_{Med} > Pr(Med^b)^{As^U} \geq Rmin_{Med}$ y $Pr(Me^b)^{As^U} == 0$ entonces esta información será utilizada para generar Bo con el mismo porcentaje de datos ME y MDE, es decir $B_0 \leftarrow ME \cup MDE$.

Si $Pr(Me^b)^{As^U} == 1$ entonces esta información será utilizada para generar B_0 unicamente con los datos ME, es decir $B_0 \leftarrow ME$.

Si $Pr(Med^b)^{As^U} == 1$ entonces esta información será utilizada para generar Bo únicamente con los datos MED, es decir $B_0 \leftarrow MED$.

En el caso de que $Rmax_{Me} > Pr(Me^b)^{As^U} \geq Rmin_{Me}$, $Rmax_{Med} > Pr(Med^b)^{As^U} \geq Rmin_{Med}$ y $Pr(Me^b)^{As^U} > 0$ entonces esta información será utilizada para generar B_0 con el mismo porcentaje de datos ME, MDE y NE, es decir $B_0 \leftarrow ME \cup MDE \cup NE$.

En la generación de población inicial B_0 el tamaño de la cadena de anticuerpos será binaria, por lo que es la cadena con un tamaño mínimo de tareas que pueden presentar información para la detección. Para determinar el número de detectores de n_D en el espacio Bo es necesario calcular el número de combinaciones con la siguiente ecuación $n_D = \frac{c!}{(c-2)!}$, donde c= number of elements of tasks S_{An}.

Fase 2: Definición de la porción de espacio normal (itself)

En esta fase compara los anticuerpos generados en la fase anterior con secuencias históricas Hs^U utilizando el algoritmo KMP. En este proceso se determinan anticuerpos con comportamientos normales, es decir, se examina uno por uno la secuencias binarias de tareas Tsk_n generadas en la poblacion inicial B_0 con toda la cadena de tareas de S^U existentes en Hs^U, comparadolas con cada detector de D_n. Si dentro de las sesiones Hs^U se determina que existe un detector de D_n este detector será marcado como normal, de lo contrario se identifica como D_{an} anormal. Con los detectores normales identificados continua a la siguiente fase.

Fase 3: La eliminación de todos los detectores que se solapan en la región normal (itself)

En esta fase se eliminan todos los detectores identificados como normales Dn y detectores con un comportamiento anormal D_{an} se conservan. Finalmente los detectores anómalos continuan al siguiente paso del sistema de detección, para identificar anomaías en secuencias activas As^U. El algoritmo de selección negativa que se utiliza en este trabajo emplea la función de emparejamiento r-contigua en una sesión activa As^U del usuario. En el experimento, el umbral de coincidencia se define como un número de detectores que se fijan dependiendo del error de falsos negativos generados en el entrenamiento, como se muestra en el trabajo de [10].

Paso 4. Detección de secuencias anómalas En este último apartado, se analiza la secuencia activa As^U y se compara con todo el conjunto de detectores anómalos D_{an} aplicando el algoritmo KPM con un umbral de estimulación definido como Sth. Si Sth es igual a TRUE el detector se activa y el antígeno lo clasifica como anómalo, de lo contrario, se clasifica como normal.

5 Resultados

Para la prueba del algoritmo de detección de anomalías se utiliza la información diez usuarios de un sistema informático real. Durante el estudio se observó que un factor principal para el óptimo funcionamiento de los algoritmos de AIS

es la estructura correcta de los datos reales de comportamiento del usuario. Estructura incorrecta impide la detección de algunos comportamientos anómalos la diferencia entre la secuencia de datos con y sin anomalías puede ser muy sutil y el algoritmo no será capaz de detectarlos. Los resultados de las pruebas de la aplicación del algoritmo AIS muestran la tasa de Clasificados Correctamente (CC), tasa Clasificados Incorrectamente (IC). También presenta la detección Tiempo promedio (segundos) para cada usuario del estudio. Estos resultados se ilustran en la tabla 1.

USR	**CC %**	**IC %**	**Tiempo (s)**
1	96.945	3.055	71.23
2	95.697	4.303	81.56
3	96.276	3.724	65.98
4	97.318	2.682	77.37
5	94.878	5.122	76.82
6	95.192	4.808	68.84
7	96.062	3.938	69.56
8	95.71	4.29	82.55
9	96.198	3.802	74.88
10	95.058	4.942	86.41
Media	**95.927**	**3.911**	**74.995**

Table 1. Tasa de detección y tiempo de detección para cada usuario utilizando el algoritmo AIS

Detección e identificación de comportamiento anómalo y conductas anómalas pueden generalizarse como lo siguiente: verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN).

En la actualidad, el sistema de detección de anomalías requiere una precisón de la clasificación, tasa de detección y tasa de falsas alarmas para evaluar el desempaño del algoritmo. Este análisis de detección de anomalías se ilustran en la tabla 2.

Finalmente, los resultados medios obtenidos a través de este trabajo son buenos con una tasa clasificado correctamente alta y podemos ver que la precisión (95,928%), la tasa de detección (91,422%) y la tasa de falsas alarmas son inferiores a 2,483%, lo que significa que la detección tiene identificación aceptable. El coste computacional es uno de el factor más importante a considerar, tanto en el entrenamiento del algoritmo y en el proceso de la detección del comportamiento del usuario. El tiempo promedio para detectar es muy alto y este es el gran problema de aplicar en un sistema informático real. El costo del entranemiento no es un factor de riesgo porque el entrenamiento se realiza sólo una vez en cada sistema en la fase inicial. El costo en la detección es un factor de riesgo para AIS, como la detección se realiza varias veces mientras el usuario ejecuta tareas en el sistema informático.

USR	TP	TN	FP	FN	Precisión	%Detección	False Alarms
1	689	1215	45	15	96.945	97.869	3.571
2	448	1309	56	23	95.697	95.117	4.102
3	406	1585	36	41	96.276	90.828	2.220
4	539	1965	30	39	97.318	93.252	1.503
5	319	1515	52	47	94.878	87.158	3.318
6	332	1628	53	46	95.192	87.8307	3.1528
7	497	1552	36	48	96.062	91.193	2.267
8	471	1537	36	54	95.710	89.714	2.289
9	557	1594	31	54	96.198	91.162	1.908
10	644	1241	35	63	95.058	91.089	2.743
				Media	**95.928**	**91.422**	**2.483**

Table 2. Resultados de la detección de comportamiento anómalo de los usuarios con el algoritmo de AIS.

6 Conclusiones y trabajos futuros

Este artículo se ha propuesto y validado un sistema inmune Artificial (AIS) con un enfoque basado en la detección de anomalías en el sistema informático inspirado en la selección negativa de las secuencias de tareas del usuario. El AIS se aplicó para comprobar el análisis de sensibilidad y definir el mejor rendimiento de parámetros de detección. La eficiencia de la AIS propuesto, que se muestra en la tabla 1 y 2, es notable y capaz de proporcionar una mejor tasa de detección con el mismo o en algunos casos menor tasa de falsas alarmas para varias tareas ejecutadas en el sistema informático. Los resultados presentan que el AIS propuesto automatiza y mejora la etapa de detección de anomalías con la aplicación de selección negativa en tareas específicas para generar secuencias de tareas anómalas. Esta forma ofrece un mejor equilibrio entre la tasa de detección y tasa de falsas alarmas, también comprueba su adaptabilidad al comportamiento humano para la detección aplicando el enfoque propuesto.

References

1. Tizard, I. R. (1992). Immunology, an introduction. Saunders College Publishing.
2. Dasgupta, D. (1999). An overview of artificial immune systems and their applications. In Artificial immune systems and their applications (pp. 3-21). Springer Berlin Heidelberg.
3. Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., and Srivastava, J. (2003, May). A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. In SDM (pp. 25-36).
4. Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 15.
5. Warrender, C., Forrest, S., and Pearlmutter, B. (1999). Detecting intrusions using system calls: Alternative data models. In Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on (pp. 133-145). IEEE.

6. Gao, B., Ma, H. Y., and Yang, Y. H. (2002). Hmms (hidden markov models) based on anomaly intrusion detection method. In Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on (Vol. 1, pp. 381-385). IEEE.
7. Helman, P., and Bhangoo, J. (1997). A statistically based system for prioritizing information exploration under uncertainty. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 27(4), 449-466.
8. Javitz, H. S., Valdes, A., and NRaD, C. (1993). The NIDES statistical component: Description and justification. Contract, 39(92-C), 0015.
9. Mykerjee, B. L. Heberlein T., and K. Levitt N.,". Network Intrusion Detection, 14-26.
10. Forrest, S., Javornik, B., Smith, R. E., and Perelson, A. S. (1993). Using genetic algorithms to explore pattern recognition in the immune system. Evolutionary computation, 1(3), 191-211.
11. Gawrychowski, P., Jez, A., & Jez, L. (2014). Validating the Knuth-Morris-Pratt failure function, fast and online. Theory of Computing Systems, 54(2), 337-372.
12. Li, Dong, Shulin Liu, and Hongli Zhang. "Negative selection algorithm with constant detectors for anomaly detection." Applied Soft Computing 36 (2015): 618-632.