ОПРЕДЕЛЯНЕ НА ФУКЦИЯТА НА ПРИГОДНОСТ В ГЕНЕТИЧНИЯ АЛГОРИТЪМ ЗА ОТКРИВАНЕ НА ПРОНИКВАНЕ В КОМПЮТЪРНИТЕ МРЕЖИ

Андон Лазаров, Петя Петрова

FITNESS FUNCTION DEFINITION IN GENETIC ALGORITHM FOR COMPUTER NETWORK INTRUSION DETECTION

Andon Lazarov, Petia Petrova

***Abstract*:** The present work suggests a thematic survey of Fitness functions' main definitions used to evaluate of network characteristics and to detect computer network intrusions. Three methods of Fitness function's calculations are considered and systemized. A genetic algorithm for training and intrusion detection based on R. H. Gong's method is described. The accent is made on Firas Alabsi's method to define a Fitness function of network characteristics. Its application is illustrated by experimental simulated data of different network connections, Normal, DoS, R2L, U2R, and Probe.

Key words: Genetic algorithm, Fitness function, network intrusion detection

1. Въведение

Генетичният алгоритъм е инструмент за търсене и намиране на оптимални решения, базирани на евристичния подход и еволюционната теория в природата с типичните характеристики, рекомбинация, наследяване, мутация [1]. Специфичната структура и основни характеристики правят генетичните алгоритми пригодни за откриване на злонамерени прониквания в компютърните мрежи. Последователността от мрежови характеристики (IP адреси, Port адреси, времетраене, протоколи и т.н.) или атрибути, заемащи определени позиции при комуникационния обмен (свързаност в компютърната мрежа) се интерпретира като хромозома, а съставните компоненти като гени [2]. Гените на всяка хромозома се кодират със символи, десетични, шестнадесетични или двоични числа. При откриване на проникване в компютърната мрежа, характеристиките на хромозомата са проблемът, който трябва да бъде решен. Те дефинират правило, което да бъде използвано за откриване на проникване и съхранено в база от данни (знания). Съвкупност от хромозоми по време на еволюционния процес се нарича популация. По време на еволюционния процес операторите рекомбинация и мутация, се използват за симулиране на естественото възпроизвеждане и мутацията на генерираните правила. За оценка на пригодността на хромозомата (правилото) за откриване на проникване в компютърната мрежа се въвежда целева функция, наречена Fitness функция или функция на пригодност.

Елементарна структура на Fitness функция, използвана в генетичен алгоритъм за откриване на smurf (DOS) и probe атаки е показана в [3)]. Оригинална Fitness функция, използваща техника на възнаграждение-наказание за ефективна оценка на хромозомите на популацията е предложена в [4]. Прилажение на генетичен алгоритъм за система за откриване на прониквания в мрежите с оценка на правилата за откриане чрез различни по дефиниция Fitness функции е показано в [5]. Аналитичен израз за изчисление на Fitness функция за оценка на пригодността на дадено правило (хромозома) е представен в [6].

Целта на настоящото изследване е да се направи тематичен обзор и класификация на известни методи за определяне на функцията на пригодност на правило (хромозома), изведено чрез генетичен алгоритъм, дефиниране на основните компоненти на Fitness функцията и експериментална оценка на пет категории на прониквания, Normal, DoS, R2L, U2R, Probe чрез алгоритъма на Firas Alabsi.

Съдържанието на статията е организирано по следния начин. Във 2 част се привежда структура на хромозомата (правилото) и описват алгоритмите за изчисление на Fitness функция за оценка на хромозомите в генетичния алгоритъм за откриване на проникване и изграждане на база от правила (знания). В 3 част са описани основните операции върху хромозомите от дадено поколение и е направена експериментална оценка на Fitness функция на Firas Alabsi. В 4 част са

формулирани общи изводи и заключение и дадени насоки за бъдещи изследвания в областта на откриване и превенция на проникванията в компютърните мрежи.

2. Структура на хромозомата (правилото) и алгоритми за изчисление на Fitness функция на хромозома в генетичния алгоритъм

Всяко правило за откриване на проникване в компютърната мрежа е клауза *if-then*, която формира следната структура на хромозомата

if A *then* B,

където A дефинира условната част на правилото (*condition*), която включва всички мрежови характеристики (гени на хромозомата), B дефинира реакцията (*act*) (дефиницията на проникването, решението или действието)
Генетичният алгоритъм за идентифициране на проникване и типа на атаки в компютърни мрежи в съответствие с литературните източници използва следните алгоритми за изчисление на типове целеви функции на пригодност за оценка на мрежовите характеристики, дефиниращи правилата (хромозомите) за идентификация на проникването.

Алгоритъм на Vrishali Yewale [5]

Алгоритъмът на Vrishali Yewale включва следните стъпки за изчисляване на Fitness функцията:
Първо, изчислява се outcome (резултатът), като сума от произведенията на съвпадението (matched) на полето A на условието “*if*” на атакуваните свързвания с полето на A на предварително класифицирания набор от данни, умножено с теглото на това поле, стойността на съвпадение (matched) е 0 или 1.

$$Outcome = \sum_i Matched * Weight, \quad (1)$$

където i = 1-57 е броят на характеристиките (полетата в хромозомата)
Стойностите теглата във функцията са в следния низходящ ред [7]:

weight(Destination IP address) > weight(Source IP address) > weight(Destination Port Number) > weight(Duration) > weight(Bytes sent by the Originator) > weight(Bytes sent by the Receiver) > weight(State) > weight(Protocol) > weight(Source Port Number)

Стойностите на теглата са класирани според позициите на полетата в записа на мрежовите характеристики, регистрирани от мрежовите регистратори. Следователно, всички гени, представляващи полето на IP адреса на дестинацията, имат едно и също тегло. Действителните стойности могат да бъдат фино настроени по време на изпълнение. Основната идея на този ред от тегла е стойността на различните полета в TCP/IP пакети. IP адрес на дестинацията (Destination) е целта на проникването, докато IP адресът на източника е инициаторът (началото – originator) на проникването. Това е най-важната информация, необходима за улавяне на проникване. Номерът на целевия Destination порт указва на приложенията, на които атакуваната система работи (например, FTP приложение обикновено се изпълнява на порт 21, HTTP приложение се изпълнява на порт 80). Някои IP адреси са по-вероятни цели за проникване - например IP адреси домейни на институции на националната сигурност. Домейн-специфичната информация е по-малко важна в сравнение с изходните IP адреси. Други параметри като продължителност, байтове, изпратени от реализаторът на проникването, байтове, изпратени от получателя, и състояние обикновено са по-малко важни от горните полета, но все още са полезни. Полетата на номер на порт и на протокола и на източника обикновено са незадължителни и се използват за определяне на специфични проникванията.
Второ, изчислява се абсолютната стойност на разлика между резултата (1) на хромозомата и действителното ниво на подозрение.

$$\Delta = \textbf{mod}(Outcome - Suspicious\ Level) \quad (2)$$

Нивото на подозрение (*Suspicious Level*) е праг, който показва степента, до която две мрежови последователности от характеристики се считат за близки, т.е. има съвпадение (matched). Действителната стойност на нивото на подозрение е величина, показваща степен на съответствие на наблюдаваните стойности на мрежовите характеристики с тези от базата с данни.

Трето, изчислява се наказателната стойност, ако се случи несъответствие, т.е.

$$Penalty = (\Delta * Ranking)/100, \quad (3)$$

Ranking в израза (3) е реципрочна стойност на вероятност за идентифициране на проникването.
Четвърто, изчислява се стойността на Fitness функцията. Стойностите на Fitness функцията са в интервала [0,1].

$$Fitness = 1 - Penalty \quad (4)$$

Алгоритъм на Ren Hui Gong [8]
Алгоритъм на Ren Hui Gong използва *Support-Confidence* Framework за идентифициране на мрежовите прониквания или точно класифициране на типа на проникване.

$$\begin{aligned} Support &= |A\ and\ B| / N \\ Confidence &= |A\ and\ B|/|A| \\ Fitness &= w_1 * support + w_2 * confidence \end{aligned} \quad (5)$$

където N е общият брой на мрежовите връзки в проверяваните данни, |A| означава броя на мрежовите връзки, отговарящи на условието A, |A и B| е броят на мрежовите връзки, които съответстват на правило *ако* A *тогава* B (*if* A *than* B). Теглата w_1 и w_2 се използват за контрол на баланса между *Support* и *Confidence*, така, че изменението на Fitness функцията е в интервала [0,1].

Генетичен алгоритъм на Ren Hui Gong за откриване и процес на обучение

Входни данни: Мрежови данни за проверка, брой на генерации, размер на популация.
Изходни данни: Множество от класификационни правила (a set of classification rules).

1. Инициализация на популацията (Initialize the population)
2. W1 = 0.2, W2 = 0.8, T (threshold) = 0.5
3. N = total number of records in the training set
4. *For* each chromosome in the population
5. A = 0, AB = 0
6. *For* each record in the training set
7. *If* the record matches the chromosome
8. AB = AB + 1
9. end *if*
10. *If* the record matches only the “condition” part
11. A = A + 1
12. end *if*
13. end *for*
14. Fitness = W1 * AB / N + W2 * AB / A
15. *if* Fitness > T
16. Select the chromosome into new population
17. end *if*
18. end *for*

19. *For* each chromosome in the new population:
20. Apply *crossover operator* to chromosomes
21. Apply mutation operator to chromosome
22. end *for*
23. If N is not reached go to line 4.

Генетичният алгоритъм стартира с произволно множество от хромозоми (ред 1). Теглата и праговите стойности за Fitness се инициализират в ред 2. Линия 3 изчислява общия брой на записите от мрежови прониквания в базата от данни. Редове 4-18 изчисляват пригодността на всяко правило (хромозома) и избират най-подходящите правила в новата популация. Редове 19-22 прилагат операторите на кръстосване и мутации към всяко правило в новата популация. На ред 23 се проверява и решава дали да прекрати процеса на сравняване или да влезе в следващото поколение, за да продължи процеса на еволюция [8].

Алгоритъм на Firas Alabsi [4]

Алгоритъм на Firas Alabsi за изчисление на Fitness функция се основана на концепцията Reward Penalty (награда - наказание) [6]. Хромозомите (правилата) се различават по силата и слабостта си (strength и weakness). Следователно, Fitness функцията отчита, първо, награда (Award), която трябва да бъде толкова, колкото силата на хромозомата и, второ, наказание (Penalty), което трябва да бъде толкова, колкото хромозомната слабост.
Прониквания в компютърната мрежа се класифицират в пет категории: Normal, DoS, Probe, U2R и R2L [9]. Всеки запис за категория в базата от данни се сравнява с данните, които постъпват в мрежата. Използват се пет характеристики (гени) за сравнение на хромозомите във всяка категория на проникване. Оригиналното описание на петте основни характеристики за всяка категория е както следва [10].

Normal:
 hot indicators: Number of “hot” indicators
 destination bytes: Number of bytes sent from the destination system to the host system
 source bytes: Number of bytes sent from the host system to the destination system
 compromised conditions: Number of compromised conditions
 dst_host_rerror_rate: % of connections that have REJ errors from a destination host

Probe
 dst_host_diff_srv_rate: % of connections to different services from a destination host
 rerror_rate: % of connections that have REJ errors
 srv_diff_host_rate: % of connections that have same service to different hosts
 logged in: binary decision
 service: type of service

DoS
 count: Number of connections made to the same host system in a given interval of time
 compromised conditions: Number of compromised conditions
 wrong_fragments: no of wrong fragments
 land: 1 if connection is from/to the same host/port; 0 otherwise
 logged in: 1 if successfully logged in; 0 otherwise

U2R
 root shell: 1 if root shell is obtained; 0 otherwise
 dst_host_srv_serror_rate: % of connections to the same service that have SYN errors from a destination host
 no of file creations: no of file creation operations

 serror_rate: % of connections that have SYN errors
 dst_host_same_src_port_rate: % of connections to same service ports from a destination host

R2L
 guest login: 1 if the login is a “guest’ login; 0 otherwise
 no of file access: no of operations on access control files
 destination bytes: Number of bytes sent from the destination system to the host system
 fail logins: no of failed login attempts
 logged in: binary decision

където no е съкращение на думата number (номер).
Построяват се 5 таблици, по една таблица за категория, всяка таблица има название (тип на категория) и включва три колони за параметрите A, AB и Fitness функция.
Данните в колона A и колона AB имат следната динамика на изменение на своите стойности. Например, при пет характеристики за категория DoS, всяка характеристика трябва да има стойности в определен диапазон в частта „*condition*“ на правилото или равна на специфична стойност, за да се оцени записът от постъпващи данни като DoS в частта „*act*“ на правилото. В този случаи петте характеристики получават същите стойности като запис в DoS, но все още не може да се дефинира като DoS, поради неизвестна или специфична стойност на една или повече неотчетени характеристики.
Ако стойностите на характеристиките са в частта на условието или състоянието (*condition* (**c**)) на правилото (хромозомата) и името на категорията е в часта на действие (*act* (**a**)) в правилото (хромозомата), то всеки запис в базата от данни се сравнява с всички постъпващи данни. Ако състоянието (условието) и действието на избрания запис в базата от данни са равни на състоянието (условието) и действието на сравнявания постъпващ запис, тогава това ще увеличи стойността на AB в колона AB на избрания запис с 1. В противен случай, ако състоянието на избрания запис е равно на състоянието на сравнения запис, но действията (*act*) на двата записа не съвпадат, тогава стойността на колона A на избрания запис ще се увеличи с 1. С други думи, ако **c & a** на избрания запис = **c & a** на сравнения запис, тогава AB = AB + 1. В противен случай, ако **c** на избрания запис = **c** на сравнения запис, но **a** на избрания запис ≠ **a** на сравнения запис, то A = A + 1. (**c** е условие (състояние) и **a** е действие).
Fitness функция възнаграждение-наказание (Award-Penalty) се дефинира със следните имперично изведени аналитични изрази [4]:

$$\text{Fitness} = 2 + ((AB-A)/(AB+A)) + AB/X - A/Y$$
или
$$\text{Fitness} = 2 + ((AB)/(AB+A)) - A/(AB+A) + AB/X - A/Y \quad (6)$$

където X е максималната стойност AB в популацията от хромозоми (правила), Y е максималната стойност A в популацията от хромозоми (правила).
Отношението (AB/(AB + A)) отразява силата (strength) на записа. Отношението, дефинирано с израза (A/(AB + A)) отразява слабостта (weakness) на записа. Отношението AB/X определя степента, която отразява силата на записа в зависимост от най-силния запис в популацията. AB/X = 0 в най-лошия случай (ако AB стойност = 0) и AB/X = 1 в най-добрия случай (ако AB приема най-висока стойност на AB в популацията). AB/X се добавя в дясната част на Fitness функцията, за да „възнагради“ записа. A/Y определя степен на слабост на записа по отношение на най-слабия запис в популацията. A/Y = 0 в най-добрия случай (ако стойност A = 0) и резултатът = 1 в най-лошия случай (ако A приема най-висока стойност в популацията), така че стойността на A/Y трябва да бъде извадена от функцията, за да извърши „наказание“ върху записа.
Анализ на стойността на Fitness функцията, дефинирана с израза (6) при отсъствие на коефициента 2: при запис (правило) с най-висока стойност на AB и A = 0, то Fitness = 2, от друга страна, при запис на правило с най-високата стойност на A и AB = 0, то Fitness = -2. Но, Fitness функцията трябва да оценява всеки запис (правило) с неотрицателна оценка. Това налага да се добави коефициент 2 в дясната част на израза (6), за да се получи Fitness стойност на правило,

неотрицателна, не равна на 0 в най-лошия случай и Fitness стойност, равна на 4 най-добрият случай. С това интервалът от стойности на Fitness функцията е [0,4].

3. Основни операции върху хромозомите от дадено поколение и експериментална оценка на Fitness функция на Firas Alabsi

Селекция
При генериране на всяко следващо поколение част от получената популация се избира да създаде ново поколение. Решенията за избор на индивидите (хромозомите или правилата) се извеждат чрез Fitness-базиран процес, което гарантира да бъдат избрани с висока вероятност индивиди с по-високи стойности на Fitness функция. От популацията са избират двойки хромозоми, които да бъдат родители на следваща популация, т.е. да се извърши рекомбинация на двойки хромозоми [11].

Кръстосване (рекомбинация)
Кръстосването или рекомбинацията) създава едно или повече нови поколения от родителските хромозоми, за да се получат по-добри хромозоми с високи стойности на Fitness функцията.

Мутация
Мутацията променя произволно новото потомство. Това се прави, за да се предотврати попадането на всички решения в популацията от хромозоми в локален оптимум при вземане на решение.
На Фиг. 1 е показана оценка на еволюцията на текуща генерация от хромозоми чрез Fitness функция (FF) и тестова хромозома, рекомбинация и мутация, и нова генерация от хромозоми в бинарен формат; със знак ♀ и ♂ са означени хромозоми за рекомбинация; със знак ● са означени невалидна хромозома, със знак ○ е означена хромозома без изменение, със знак ◌ е означена мутираща хромозома.

Current Generation	FF	Crossover&Mutation	New Generation
1011101010001010	0.43♀		1011101010001010
1000011110101000	0.18●	1011100100001001	1011100100001001
1100101000101011	0.13●	0101111010001010	0101111010001010
0101110100001001	0.34♂		0101110100001001
0101000111010101	0.27◌	0101000011010101	0101000011010101
0010111000101011	0.25○		0010111000101011

↑

Test Chromosome
0101111010001101

Фиг. 1. Оценка на текуща генерация от хромозоми чрез Fitness функция (FF) и Test Chromosome, рекомбинация и мутация, и нова генерация от хромозоми в бинарен формат; ♀ и ♂ хромозоми за рекомбинация; ● невалидна хромозома, ○ хромозома без изменение, ◌ мутираща хромозома.

За да се потвърди оценката на хромозомата с дадена Fitness функция, хромозомата трябва да се тества с друга Fitness функция, като резултатите върху хромозомата от двете Fitness функции се сравняват. Като втора контролна Fitness функция се използва Support Confidence Framework, дефинирана с (5), която с цел целостта на изчислението на Fitness функцията ще се представи отново с уравненията

Support = |A *and* B| / N
Confidence = |A *and* B|/|A|
Fitness = w_1 * *support* + w_2 * *confidence*,

където *Support* показва относителния брой повторението на AB в рамките на всички правила в популацията. *Confidence* показва относителния брой на AB в рамките на тези правила, които имат една и съща условна част (*condition*). w_1 и w_2 са прагови или теглови стойности за балансиране между стойността на *Support* и стойността на *Confidence*. В този случай се приема (w_1 = 0.0257) и (w_2 = 0.9843).
Ако стойността Fitness функцията на правилото (хромозомата) P е по-голяма от Fitness стойността на правилото (хромозомата) Q според първата Fitness функция, тогава стойността Fitness функцията на правилото P също е по-голяма от стойността на Fitness функцията на правилото Q според втората Fitness функция. За всеки запис (хромозома) в популацията съществуват два резултата R_1 и R_2, дефинирани със следните две уравнения

$$R_1 = \frac{Fv_1}{\max Fv_1 \, in Populatio} \tag{7}$$

$$R_2 = \frac{Fv_2}{\max Fv_2 \, in Populatio} \tag{8}$$

където Fv_1 е резултат от Fitness функцията, основана на Reward-Penalty Fitness функция, Fv_2 е резултат на Support Confidence Framework Fitness функция на същата хромозома (запис). За да се потвърди качеството на новата Fitness функция, т.е. с нея се получава добър резултат, стойностите на R_1 и R_2 трябва да бъдат близки една към друга.
Обобщен псевдо-код на базов генетичен алгоритъм [12]:

```
1. InitPopulation(P)
2. Fitness(P)
3. while MaxGenerationNotReached do
4. for i = 0 to xfactor do
5. p1 = Selection(P)
6. p2 = Selection(P)
7. (o1, o2) = crossover(p1, p2)
8. crowding(p1, p2, o1, o2)
9. end for
10. for i = 0 to dfactor do
11. p = Selection(P)
12. Dropping(p)
13. end for
14. for i = 0 to mfactor do
15. p = Selection(P)
16. Mutation(p)
17. end for
18. Fitness(P)
19. end while
20. SelectBestIndividual(P)
```

където x означава crossover - кръстосване (рекомбинация) на хромозоми, d означава dropping - отпадне на хромозома, m означава mutation - мутация на хромозома

Оценки на A, AB, и Fitness функция с прилагане на Fitness функция на Firas Alabsi, получени с данни от симулационен експеримент на Normal, DoS, R2L, U2R, Probe мрежови комуникации, реализирани с случайно генерирани мрежови характеристики на пет хромозомни структури за всяка категория, са представени в Таблица 2.

Normal		
A	AB	Fitness
6	5	1.91
0	12	3.21
69	3	0.136
2	56	3.9
1	27	3.39

DoS		
A	AB	Fitness
7195	31	0.0921
349	371	2.982
4	9	2.408
12	54	2.78
87	226	3.041
R2L		
A	AB	Fitness
5	21	2.641
1	7	2.75
148221	19	0.0235
513	12	1.057
19	817	3.954
U2R		
A	AB	Fitness
14	2	1.707
3	56	3.898
71	9	1.385
129101	4	0.0715
25	16	2.066
Probe		
A	AB	Fitness
84	915	3.831
92	18	1.34
9	4	1.62
141802	11	0.0122
3	28	2.837

Таблица 2: Стойности на A, AB и Fitness функция на експериментални записи (хромозоми) на Normal, DoS, R2L, U2R, Probe сиулирани мрежови комуникации.

Резултатите от таблица 1 могат да бъдат анализирани както следва. Максимална стойност на Fitness функцията на хромозома от категории Normal, R2L, U2R, Probe се полуава при максимална стойност на AB, което показва максимално съвпадение на двете части на клаузата *if* A *than* B, както на текущата последователност от мрежови характеристики (сравняваната хромозома), така и на последователност от мрежови характеристики на хромозомата от базата с

данни с априорно дефинирана категория. Минимална стойност на Fitness функцията на хромозома от петте категории се полуава при максимална стойност на A, което показва максимално съвпадение само на A част на клаузата *if* A *than* B на текущата последователност от мрежови характеристики (сравняваната хромозома), и на последователност от мрежови характеристики на хромозомата от базата с данни с априорно дефинирана категория. Експерименталните резултати за категория DoS показват, че максимална стойност на Fitness функцията се получава при AB = 226, а не при максимална стойност на AB, за която Fitness функцията е 2.982. Това е в резултат на високата стойност на A = 349, която съществено увеличава „наказанието“ върху стойността на Fitness функцията.

4. Заключение

В настоящата статия е направен целеви и систематизиран обзор на методите за изчисление на функцията на пригодност (Fitness function) оценка на мрежовите характеристики (гени) при откриване на проникване в компютърните мрежи, базирано на генетичен алгоритъм. Приведени са обобщен генетичен алгоритъм и псевдо-код. Направена експериментална оценка на метода на Firas Alabsi за изчисление на функцията на пригодност на генерираните правила (хромозоми).
Идеята за прилагане на генетични алгоритми в системите за откриване и превенция на проникванията в компютърните мрежи може да се развие в посока създаване на бази от данни за комуникационните характеристики на различни типове компютърни атаки. За оценка на текущи и нови проникванията се предвижда използване на интегрирани и взаимно-допълващи се оценъчни функции на пригодност, с което ще се повиши точността на оценките и ефективността на превенция на проникванията в компютърните мрежи.

Литература

1. Sharmila Devi, Ritu Nagpal. Intrusion Detection System using genetic algorithm -A Review, International Journal of Computing & Business Research, Proceedings of ‘I-Society 2012’ at GKU, Talwandi Sabo Bathinda (Punjab). ISSN (Online): 2229-6166
2. Wei Li, Using Genetic Algorithm for Network Intrusion Detection, https://pdfs.semanticscholar.org/9175/54c7cce69e6ee9708020863f2bd27fa986a6.pdf
3. Anup Goya, Chetan Kumar. GA-NIDS: A genetic algorithm based network Intrusion Detection System, 2007. https://pdfs.semanticscholar.org/6c8e/6708a1a737a9a5509de2fba46f8de1aff7e3.pdf
4. Firas Alabsi, Reyadh Naoum, Fitness function for genetic algorithm used in Intrusion Detection System, International Journal of Applied Science and Technology, Vol. 2 No. 4; April 2012, pp. 129-134.
5. Vrishali Yewale, Vimla Jethani, Tushar Ghorpade. Applying Genetic Algorithm to Intrusion Detection System, International Journal of Science and Research (IJSR), Volume 4 Issue 4, April 2015, pp. 524-529.
6. Pedro A. Diaz-Gomez, Dean F. Hougen. Improved off-line intrusion detection using genetic algorithm, in Proceedings of the Seventh International Conference on Enterprise Information Systems, 2005. http://www.cameron.edu/~pdiaz-go/Art_ICEIS.pdf
7. Wei Li. Using Genetic Algorithm for Network Intrusion Detection, In proceedings of the United States Department of Energy, 2004. https://pdfs.semanticscholar.org/9175/54c7cce69e6ee9708020863f2bd27fa986a6.pdf
8. R. H. Gong, M. Zulkernine, P. Abolmaesumi, ”A software implementation of a genetic algorithm based approach to network intrusion detection”, in Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks, SNPD/SAWN’05), 0-7695-2294-7/05, 2005.
9. An. Goya, Ch. Kumar. GA-NIDS: A Genetic Algorithm based Network Intrusion Detection System, 2007.

https://pdfs.semanticscholar.org/6c8e/6708a1a737a9a5509de2fba46f8de1aff7e3.pdf
10. S. Mukkamala, A. Sung, A. Abrham. (2004), Modeling Intrusion Detection System using Linear Genetic Programming Approach, Proceeding IEA/AIE 17th International Conference on Innovations in Applied Artificial Intelligence, pp. 633-642, ISBN: 3-540-22007-0.
https://www.researchgate.net/publication/221049814_Modeling_Intrusion_Detection_Systems_Using_Linear_Genetic_Programming_Approach
http://www.rmltech.com/doclink/LGP%20Based%20IDS.pdf.
11. Omprakash Chandrakar, Rekha Singh, Lal Bihari Barik. Application of Genetic Algorithm in Intrusion Detection System, Control Theory and Informatics, Vol.4, No.1, 2014, pp. 50-57.
12. V. Bapuji, R. N. Kumar, A. Goverdan, and S. Sharma, “Soft Computing and Artificial Intelligence Techniques for Intrusion Detection System,” Networks and Complex Systems, vol. 2, no. 4, 2012.