

## МЕТОД ВИЯВЛЕННЯ КІБЕР-ЗАГРОЗ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ

*В роботі представлено метод виявлення кібер-загроз на основі еволюційних алгоритмів. Метод дозволяє забезпечити реагування на нові загрози, забезпечуючи захист комп'ютерних систем від як відомих, так і невідомих кібер-загроз. Робота системи виявлення нових загроз здійснюється на основі обробки зібраних в мережі та в комп'ютерній системі множини ознак кібер-загроз, виділення з неї підмножини таких ознак і створення таких необхідних правил, які дозволять виявити кібер-загрози. Процес використовує генетичні алгоритми для мінімізації необхідних ознак виявлення кібер-загроз, що і дозволяє ефективно використовувати наявні ресурси для захисту від кібер-загроз.*

*Ключові слова: кібер-загрози шкідливе програмне забезпечення, генетичні алгоритми, мутація, популяція, пристосованість, еволюція.*

S. LYSENKO, D. STOPCHAK, V. SAMOTES

Khmelnytskyi National University

## TECHNIQUE FOR DETECTION OF CYBERTHREAT BASED ON EVOLUTIONAL ALGORITHMS

*The purpose of this paper is to develop a method for the cyber-threats detection based on the evolutionary algorithms.*

*In this work a method for cyber threat detection based on genetic algorithms is presented. The method allows detecting both known and previously unknown threats. The usage of the genetic algorithms allowed to use them as the basis for building a method of detecting cyber threats. The method has the heuristic nature and is based on the collected data about the cyber attacks. It makes it possible to give an answer about the presence of cyber threats in the computer network against the computer systems.*

*The mechanism of threat detection system is based on collection of threat features from network or a computer systems, extracting a subset of acquired set and generation of threat detection rules. Genetic algorithms are used for the minimization of the feature set, which allows effective using of the system resources for threat detection.*

*In this article a method for the dividing the feature space for threat detection rule generation is suggested. For division of the suggested method it is necessary to generate the threat detection sub-rule for each value of the selected feature. It is suggested to use the feature with the smallest domain for generating the minimal set for rules. It is possible to select the optimal feature after all selected features which were discovered while applying the genetic algorithm. The sub-rule set is used with the aim to reduce false positive rate.*

*Developed threat detection system consists of training and detection subsystems. When some object detected as suspicious but cannot be unambiguously identified as a threat, and there is a partial feature match for a threat with no match for a benign object, this object is considered to be used for further improvement of the system on the training stage.*

*The process uses genetic algorithms to minimize the signs of detecting cyber threats, thereby reducing the resource intensity of the process of detecting cyber threats. The proposed method has demonstrated the ability to identify cyber threats with high confidence.*

*Keywords: genetic algorithms, cyber-threats, malware, cross-overer, mutation, population, adaptation, evolution.*

Кількість активних користувачів у мережах постійно зростає, тому експоненціально зростають ризики здійснення кібер-загроз щодо них. Закон Меткалфа говорить, що корисність мережі залежить від кількості підключених користувачів, і пропорційна їх числу [1]. Зрозуміло, що зручність, корисність і необхідність, в свою чергу, збільшують кількість користувачів. Проте з точки зору кібер-безпеки, кожен додатковий користувач, додаток або функція, що працюють в мережі, збільшує можливість для атаки.

Одним з підходів до виявлення кібер-загроз є залучення механізмів машинного навчання, яке дозволяє не лише виявляти, але й запобігати атакам нульового дня. Такі підходи покладаються на евристичні механізми і нечіткі відповідності, що надає перевагу перед звичайними методами, зокрема перевірки сигнатур.

Системи виявлення вторгнень розглядаються як перша лінія захисту для комп'ютерних систем. Вони призначені для моніторингу та захисту комп'ютерних систем. Системи виявлення вторгнень динамічно контролюють і аналізують події, що відбуваються в системі, і визначають ступінь їх легітимності [2].

З огляду на стан речей, що склався на даний момент, актуальною є задача розроблення методу виявлення кібер-загроз, який мав би евристичну природу, яка б забезпечувалася застосуванням апарату еволюційних алгоритмів. Виявлення нових загроз має здійснюватися на основі використання зібраних даних, а результат щодо наявності кібер-загрози отримують шляхом роботи побудованих правил. Метод повинен забезпечити реагування на нові загрози, забезпечуючи захист комп'ютерних систем від як відомих, так і невідомих кібер-загроз.

**Генетичні алгоритми як засіб виявлення кібер-загроз.** Антивірусне програмне забезпечення, яке використовує технології, що базуються на сигнатурних аналізаторах, зазвичай не може виявити шкідливе програмне забезпечення (ШПЗ) нульового дня [3]. Вирішення цієї проблеми покладено на евристичні методи виявлення. Генетичні алгоритми відносять до евристичних алгоритмів. Вони дозволяють віднайти прийнятне рішення в більшості випадків, проте при цьому правильність рішення математично не доводиться і застосовується найчастіше для задач, аналітично вирішити яких дуже важко чи неможливо. Також генетичні алгоритми дозволяють формувати правила виявлення ШПЗ на основі попередніх виявів загроз.

Еволюційні обчислення за своєю суттю є оптимізаційним методом та системою автоматизації проектування для унікального механізму відбору, оскільки він отримав це від природи, зокрема біологічної

основи [4]. В генетичних алгоритмах закладені принципи, запозичені у природи. Ці принципи - спадковість і мутація. Спадковість – здатність організмів передавати свої ознаки і особливості розвитку потомству. Завдяки цій здатності всі живі істоти зберігають в своїх нащадках характерні риси виду. Мутація генів у живих організмів забезпечує генетичну різноманітність популяції і має випадковий характер, так як природа заздалегідь не знає, які саме ознаки особин будуть найкращими в майбутньому (зміна клімату, зменшення або збільшення кількості їжі, поява конкуруючих видів і т.п.). Саме мутація дозволяє з'явитися особинам з новими ознаками, які можуть вижити і залишити потомство в нових, змінених умовах [5].

Природний відбір, схрещування і мутація забезпечують розвиток популяції. Кожне нове покоління вважається більш пристосованим, ніж попереднє. Воно краще задовольняє вимогам зовнішнього середовища. Цей процес називається еволюцією.

Процес схрещування називається кросовером і є основним генетичним оператором, за рахунок якого здійснюється обмін генетичним матеріалом між особами.

Як відомо, принцип природного відбору полягає в тому, що в конкурентній боротьбі виживає найбільш пристосований. Зазвичай пристосованість особи визначається цільовою функцією, яку називають фітнес - функцією.

Генетичні алгоритми використовуються для аналізу величезних обсягів багатовимірних даних, які обробляються системою виявлення вторгнень [6]. Генетичні алгоритми еволюціонують з часом, що робить їх гарним вибором для динамічного формування правил, які в подальшому будуть здатними розв'язувати поставлену задачу.

Таким чином, використання можливостей генетичних алгоритмів дозволить використати їх як основу для побудови методу виявлення кібер-загроз. При цьому метод матиме евристичну природу і на основі зібраних даних щодо кібер-атак буде спроможним дати відповідь щодо наявності кібер-загрози.

**Метод виявлення кібер-загроз на основі еволюційних алгоритмів.** В статті пропонується метод виявлення кібер-загроз на основі еволюційних алгоритмів. Метод дозволяє забезпечити реагування на нові загрози, забезпечуючи захист комп'ютерних систем від як відомих, так і невідомих кібер-загроз. Робота системи виявлення нових загроз здійснюється на основі обробки зібраних в мережі та в комп'ютерній системі множини ознак кібер-загроз, а також на основі виділення з неї підмножини таких ознак і створення таких необхідних правил, які дозволять виявити кібер-загрози.

Процес використовує генетичні алгоритми для мінімізації необхідних ознак виявлення кібер-загроз, що і дозволяє ефективно використовувати наявні ресурси для захисту від кібер-загроз.

Метод складається з двох етапів: 1) етап навчання системи виявлення кібер-загроз; 2) етап виявлення кібер-загроз.

Укрупнена схема функціонування методу виявлення кібер-загроз на основі еволюційних алгоритмів подана на рис. 1.

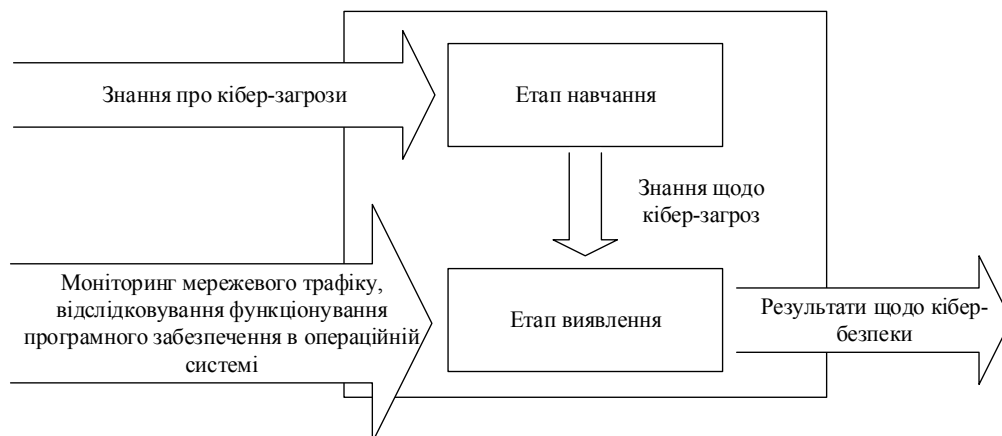


Рис. 1. Укрупнена схема функціонування методу виявлення кібер-загроз на основі еволюційних алгоритмів

Розглянемо кроки функціонування етапу навчання як складового метода виявлення кібер-загроз на основі еволюційних алгоритмів:

- 1) формування знань щодо кібер-загроз на основі відомих ознак та їх проявів;
- 2) формування вектору ознак щодо можливої кібер-загрози;
- 3) здійснення аналізу та поділ кібер-загроз класи;
- 4) застосування генетичного алгоритму для мінімізації кількості ознак, необхідних для виявлення кібер-загроз та віднесення їх до певного класу;
- 5) формування правил для виявлення кібер-загроз та віднесення їх до певного класу.

Розглянемо кроки функціонування етапу виявлення кібер-загроз на основі еволюційних алгоритмів:

- 1) збір інформації, що вказує на можливу присутність кібер-загроз (моніторинг мережевого трафіку, відслідковування функціонування програмного забезпечення в операційній системі);
- 2) здійснення виявлення кібер-загроз на основі застосування генетичного алгоритму з подальшим

віднесенням їх до певного класу;

3) на основі одержаного результату здійснення відповідного реагування на можливу кібер-загрозу.

Загальна схема функціонування методу виявлення кібер-загроз на основі еволюційних алгоритмів подана на рис. 2.

Розглянемо більш детально перший етап методу виявлення кібер-загроз на основі еволюційних алгоритмів.

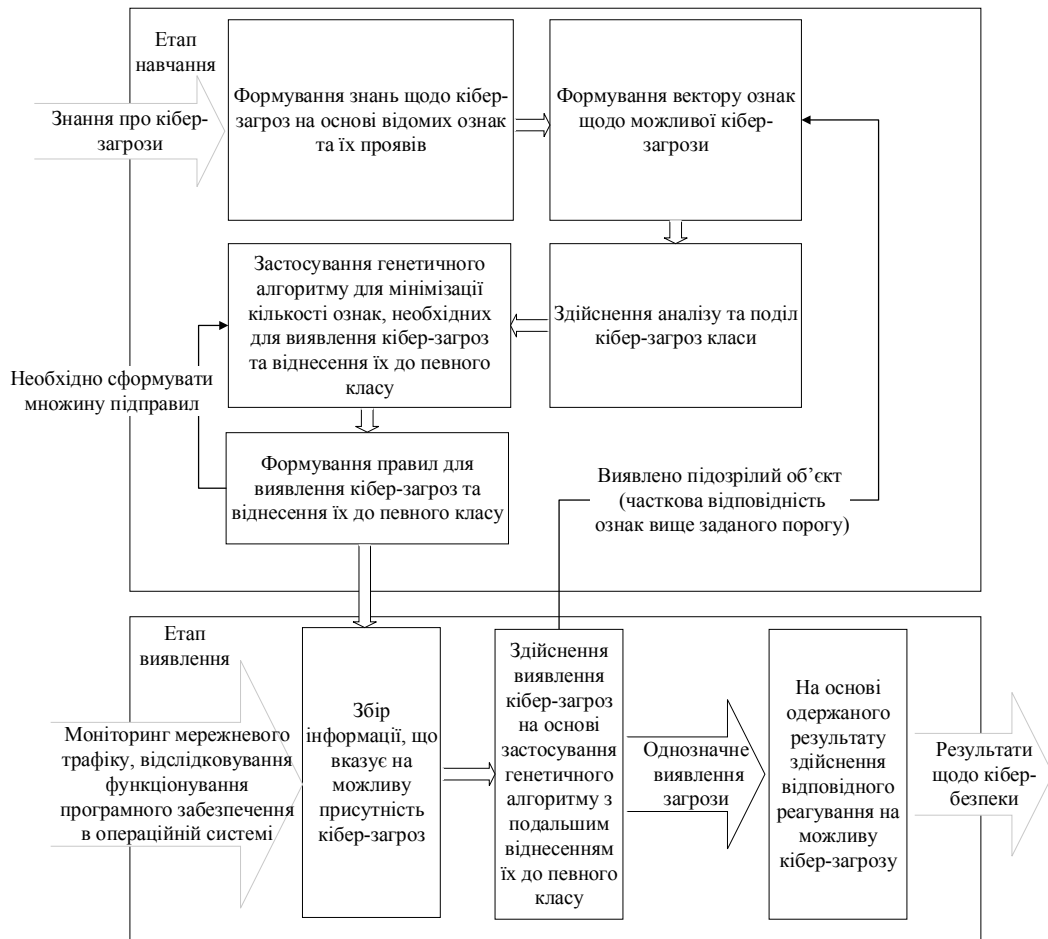


Рис. 2. Загальна схема функціонування методу виявлення кібер-загроз на основі еволюційних алгоритмів

*Формування знань щодо кібер-загроз на основі відомих ознак та їх проявів.* Для генерації правил виявлення кібер-загроз використано множини ознак (поведінка підключень, використання команд, з'єднань до служб тощо), що свідчать про їх присутність.

*Формування вектору ознак щодо можливої кібер-загрози.* Для побудови знань щодо кібер-загроз в роботі розглядається поняття «з'єднання» - послідовність TCP-пакетів, що починаються і закінчуються в визначений час, і між якими потоки даних надсилаються з IP-адреси зловмисника до цільової IP-адреси жертви в рамках визначеного протоколу. Кожне з'єднання позначене як звичайне, або як загроза з точно визначеним одним конкретним типом атаки. Кожен запис з'єднання складається приблизно з 100 байт [7].

Подамо множини з'єднань як  $V = \{v_i\}_{i=1}^{N_v}$ , де  $N_v$  – загальна кількість з'єднань. Нехай  $Q = \{q_i\}_{i=1}^{N_q}$  – множина ознак з'єднань, де  $N_q$  – загальна кількість ознак, тоді  $M = (m_{ij})_{i=1, j=1}^{N_v, N_q}$  – матриця даних, кожен рядок якої є вектором ознак.

Вектор ознак з'єднань подамо наступним чином:

$$Q_{c_i} = \begin{pmatrix} L_{c_i}^N & P_{c_i}^T & S_{c_i}^T & F_{c_i} & SB_{c_i}^N & DB_{c_i}^N & L_{c_i} \\ WF_{c_i}^N & U_{c_i}^N & H_{c_i} & FL_{c_i}^N & LI_{c_i} & NC_{c_i}^N & RS_{c_i} \\ SA_{c_i}^N & R_{c_i}^N & FC_{c_i}^N & S_{c_i}^N & AF_{c_i}^N & OC_{c_i}^N & HL_{c_i} \\ QL_{c_i} & C_{c_i}^N & SC_{c_i}^N & SR_{c_i} & SSR_{c_i} & RR_{c_i} & SRR_{c_i} \\ SR_{c_i}^{srv} & DR_{c_i}^{srv} & SDHR_{c_i} & DHC_{c_i} & DHSC_{c_i} & DHSSR_{c_i} & DHDSR_{c_i} \\ DHSSPR_{c_i} & DHSDDR_{c_i} & DHSR_{c_i} & DHSR_{c_i}^{srv} & DHRR_{c_i} & DHRR_{c_i}^{srv} & \end{pmatrix},$$

де  $L_{c_i}^N$  – довжина (кількість секунд) з'єднання;  $P_{c_i}^T$  – тип протоколу (tcp, udp);  $S_{c_i}^T$  – тип протоколу прикладного рівня (http, telnet);  $F_{c_i}$  – нормальний або помилковий статус з'єднання;  $SB_{c_i}^N$  – кількість переданих байт даних від джерела до пункту призначення;  $DB_{c_i}^N$  – кількість байт даних від пункту призначення до джерела;  $L_{c_i} - 1$ , якщо з'єднання з/до того ж хосту/порта, інакше 0;  $WF_{c_i}^N$  – кількість неправильних фрагментів;  $U_{c_i}^N$  – кількість термінових пакетів;  $H_{c_i}$  – кількість гарячих показників;  $FL_{c_i}^N$  – кількість невдалих спроб входу;  $LI_{c_i} - 1$ , якщо успішно залогінились, інакше 0;  $NC_{c_i}^N$  – кількість скомпрометованих значень;  $RS_{c_i} - 1$ , якщо отримано адміністративні права, інакше 0;  $SA_{c_i}^N - 1$ , якщо команда su root спробувана, інакше 0;  $R_{c_i}^N$  – кількість адміністративних доступів;  $FC_{c_i}^N$  – кількість операцій створення файлів;  $S_{c_i}^N$  – кількість рядків командної оболонки;  $AF_{c_i}^N$  – кількість операцій на доступ до контрольних файлів;  $OC_{c_i}^N$  – кількість вихідних команд під час ftp-сеансу;  $HL_{c_i} - 1$ , якщо логін належить до списку адміністраторів, інакше 0;  $QL_{c_i} - 1$ , якщо логін гостьовий, інакше 0;  $C_{c_i}^N$  – кількість підключень до того ж хосту як поточне з'єднання в останні дві секунди;  $SC_{c_i}^N$  – кількість підключень до того ж сервісу як поточне з'єднання в останні дві секунди;  $SR_{c_i} - \%$  з'єднань, які мають помилку SYN;  $SSR_{c_i}$  – відсоток з'єднань, які мають помилку SYN;  $RR_{c_i}$  – відсоток з'єднань, які мають помилку REJ;  $SRR_{c_i}$  – відсоток з'єднань, які мають помилку REJ;  $SR_{c_i}^{srv}$  – відсоток підключень до однієї служби;  $DR_{c_i}^{srv}$  – відсоток з'єднань до різних служб;  $SDHR_{c_i}$  – відсоток підключень до різних хостів;  $DHC_{c_i}$  – кількість хосту призначення;  $DHSC_{c_i} - SC_{c_i}^N$  для хосту призначення;  $DHSSR_{c_i} - SSR_{c_i}$  для хосту призначення;  $DHDSR_{c_i} - DR_{c_i}^{srv}$  для хосту призначення;  $DHSSPR_{c_i}$  – відсоток підключень до одного порта для хосту призначення;  $DHSDHR_{c_i} - \%$  підключень до різних портів для хосту призначення;  $DHSR_{c_i}$  – відсоток підключень до поточного хосту, які мають помилку S0;  $DHSR_{c_i}^{srv}$  – відсоток підключень до поточного хосту та вказаної служби які мають помилку S0;  $DHRR_{c_i}$  – відсоток підключень до поточного хосту, які мають помилку RST;  $DHRR_{c_i}^{srv}$  – відсоток підключень до поточного хосту та вказаної служби які мають помилку RST.

Позначимо промарковану вибірку даних, побудовану на апіорних знаннях щодо приналежності аналізованих даних до типу загрози за сукупністю ознак, як  $V_l = \{v_{li}\}_{i=1}^{N_{V_l}}$ , де  $N_{V_l}$  – кількість з'єднань в промаркованій вибірці даних, а немарковану вибірку з'єднань як  $V_u = \{v_{ui}\}_{i=1+N_{V_l}}^{N_v}$ .

Нехай  $H = \{h_i\}_{i=1}^{N_h}$  – множина наперед визначених типів з'єднань, де  $N_h = 23$  – кількість типів з'єднань, що складаються з 22 загроз:  $h_1 = \text{«ack»}$ ,  $h_2 = \text{«buffer\_overflow»}$ ,  $h_3 = \text{«ftp\_write»}$ ,  $h_4 = \text{«guess\_passwd»}$ ,  $h_5 = \text{«imap»}$ ,  $h_6 = \text{«ipsweep»}$ ,  $h_7 = \text{«land»}$ ,  $h_8 = \text{«loadmodule»}$ ,  $h_9 = \text{«multihop»}$ ,  $h_{10} = \text{«neptune»}$ ,  $h_{11} = \text{«nmap»}$ ,  $h_{12} = \text{«perl»}$ ,  $h_{13} = \text{«phf»}$ ,  $h_{14} = \text{«pod»}$ ,  $h_{15} = \text{«portsweep»}$ ,  $h_{16} = \text{«rootkit»}$ ,  $h_{17} = \text{«satan»}$ ,  $h_{18} = \text{«smurf»}$ ,  $h_{19} = \text{«spy»}$ ,  $h_{20} = \text{«teardrop»}$ ,  $h_{21} = \text{«warezclient»}$ ,  $h_{22} = \text{«warezmaster»}$ . Також в вибірці присутні звичайні з'єднання, що не становлять загрозу  $h_{23} = \text{«normal»}$ , тоді  $V_h = \{v_{hi}\}_{i=1}^{N_{V_h}}$  – множина об'єктів промаркованої вибірки, що належать визначеному класу, де  $N_{V_h}$  – кількість об'єктів в класі. Задачу визначення типу загрози можна виразити функцією  $\int_{\text{detect}}: V_{\min} \rightarrow H$ .

Також важливо зазначити, що кожна ознака з'єднання характеризується визначеною множиною значень для різних типів кібер-загроз. Наприклад, при атаці pod ознака src\_bytes, кількість байтів даних, надісланих джерелом IP-адреси, має значення 1480 і 564.

*Здійснення аналізу та поділ кібер-загроз класи.* На цьому етапі необхідним є однозначне визначення типу кібер-загрози для подальшого виокремлення підмножини ознак для формування правил виявлення атаки. Наприклад в наборі NSL-KDD присутні двадцять два класи атак в навчальних даних, чотири типи:

- 1) DOS – відмова в обслуговуванні;
- 2) R2L – неавторизований доступ із віддаленого комп'ютера;
- 3) U2R – несанкціонований доступ до локальних прав адміністратора, наприклад, різні атаки «буферного переповнення»;
- 4) Зондування – спостереження, наприклад, сканування порту.

Застосування генетичного алгоритму для мінімізації кількості ознак, необхідних для виявлення кібер-загроз та віднесення їх до певного класу. Генетичний алгоритм вибирає кращих представників попереднього покоління, схрещує їх і отримує безліч нових особин. Це нове покоління містить краще співвідношення характеристик, якими володіють хороші члени попереднього покоління. Таким чином, з покоління в покоління, характеристики з кращими досягнутими показниками поширюються по всій популяції. Схрещування найбільш пристосованих особин призводить до того, що досліджуються найбільш перспективні ділянки простору пошуку. В кінцевому результаті, популяція буде сходитися до оптимального розв'язання задачі.

Для виявлення загроз розглядатимемо не тільки дискретні значення, а значення, що набувають в інтервалі. В специфіці кібер-загроз це дозволить використовувати толерантність до неточностей, невизначеності, можливої часткової істини. Це дозволить виявляти загрози, знаходячи відхилення від нормальної поведінки і ознак в даних. Якщо відхилення перевищують заданий поріг, то дані вважаються аномальними.

В роботі пропонується використання генетичних алгоритмів для створення сукупності ознак, які дозволяють виявляти кібер-загрозу. Вказані ознаки представляються у вигляді закодованих двійкових послідовностей, де 1 в розряді такої послідовності вказує на необхідність цієї ознаки для детектування визначеної загрози.

Схема функціонування генетичного алгоритму показані в на рисунку 3. Кроки роботи алгоритму такі:

1. Генерація початкової популяції з  $n$  хромосом.
2. Обчислення пристосованості для кожної особи.
3. Вибір пари осіб-батьків за допомогою одного з методів відбору.
4. Проведення кросинговеру двох предків з ймовірністю  $p_c$  для створення нащадків.
5. Проведення мутації нащадків з ймовірністю  $p_m$ .
6. Повторення кроків 3-5, поки не буде згенеровано нове покоління популяції, що містить  $n$  хромосом.
7. Повторення кроків 2-6, поки не буде досягнутий критерій закінчення процесу.

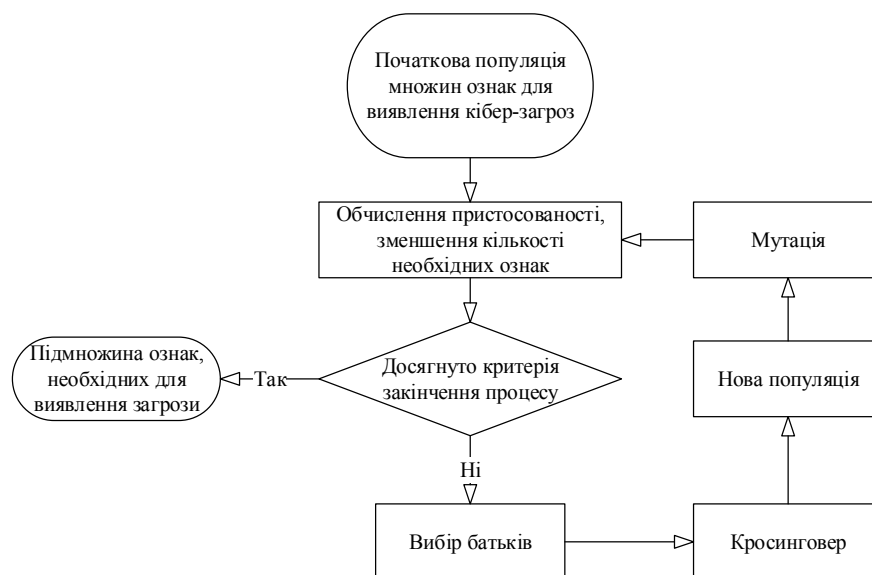


Рис. 3. Схема функціонування кроку застосування генетичного алгоритму для мінімізації кількості ознак, необхідних для виявлення кібер-загроз та віднесення їх до певного класу

Першим кроком є вибір популяції хромосом. В роботі пропонується початкову популяцію генерувати з вказанням необхідності всіх ознак для виявлення кібер-загроз. В ході роботи алгоритму залишаться тільки необхідні гени. Кожна хромосома складається з кінцевого числа генів, яка визначена в кожній реалізації. Ці хромосоми є даними, що представляють сукупність ознак, необхідних для виявлення кібер-загроз.

Другим кроком є кросинговер хромосом для появи наступного покоління. Відповідно до вимог потрібного рішення, різні позиції генів у хромосомі кодуються як цифри, біти або символи.

Третім кроком є використання мутації для вдосконалення популяції, де кожна хромосома

задовольняє заздалегідь визначену функцію фітнесу.

Четвертим кроком є перевірка досягнення поставленої мети.

Для методу виявлення кібер-загроз на основі еволюційних алгоритмів було використано 41 ознаку, які вказують на особливості функціонування з'єднань. Зокрема, було залучено мережні протоколи (TCP, UDP і т.д.), тривалість (кількість секунд) з'єднання, тип протоколу прикладного рівня (HTTP, Telnet), кількість переданих даних (байт даних від джерела до місця призначення), кількість отриманих даних (кількість байт даних від одержувача до джерела), нормальний або помилковий статус з'єднання та ін.

Для виявлення загрози не всі ознаки можуть бути необхідні, до того ж, перевірка кожної ознаки знизить ефективність роботи методу. Тому за допомогою генетичних алгоритмів необхідно виявити підмножину з наявних, які є достатніми для однозначного виявлення.

З цією метою необхідно виконати наступні кроки:

1. Здійснити кодування: гени виражаються як двійковий код. Хромосома виражається як двійковий рядок

$A_i, a_1 \dots a_n$ , де  $A_i=0,1$  і вказує на те, чи є ознака присутня в підмножині ознак для генерації правил.

2. Використати функцію фітнесу (зрозуміло, що для виявлення загрози, необхідна присутність хоча б однієї ознаки):

$$f = \frac{n - \sum_{i=1}^n a_i}{n} = 1 - \frac{1}{n} \sum_{i=1}^n a_i,$$

де  $n$  – кількість ознак.

З межами  $[0; 1)$ , де наближення до 1 – покращення результату, тобто зменшення кількості ознак, необхідних для виявлення загрози.

*Формування правил для виявлення кібер-загроз та віднесення їх до певного класу.* Цей крок передбачає формування необхідних підмножин ознак, достатніх для виявлення кібер-загроз. За допомогою методу виявлення кібер-загроз на основі еволюційних алгоритмів здійснюється побудова множини правил на їх основі, для виявлення однієї загрози можливе створення декількох правил з різними підмножинами ознак. Можливе використання ознак нормальних з'єднань для вдосконалення методу і виключення хибних реагувань.

Якщо правило не покриває необхідну кількість виявлених кібер-загроз (наприклад, не менше 99%), або присутня велика кількість помилок другого роду, на етапі формування правил для виявлення кібер-загроз та віднесення їх до певного класу, то метод вибирає ознаку, яка може приймати більш ніж одне значення. Після цього здійснюється повернення до попереднього кроку (застосування генетичного алгоритму для мінімізації кількості ознак, необхідних для виявлення кібер-загроз та віднесення їх до певного класу) з метою формування правила для кожного можливого значення вибраної ознаки. Вибір ознаки здійснюється послідовно. З метою генерації меншої кількості необхідних правил для виявлення загроз можливий вибір ознак з мінімальним можливим набором значень. Результатом роботи етапу «Формування правил для виявлення кібер-загроз та віднесення їх до певного класу» виступає правило або сукупність правил для виявлення кібер-загроз з кількістю, що дорівнює кількості унікальних значень ознаки, що була вибрана для поділу, або більше, в разі подальшого поділу.

Прийmemo  $G$  як множину усіх правил. Нехай  $g_i$  – одне правило з множини.

$g_i = (A_1) \wedge \dots \wedge (A_n)$ , де  $n$  – кількість ознак, необхідних для виявлення кібер-загрози,  $A_n$  – ознака з сукупністю значень кібер-загрози,  $B_n$  – ознака з сукупністю значень для нормального з'єднання.

$A_k = \{a_{k_i}\}_{i=1}^{N_{A_k}}$ , де  $k$  – кількість ознак кібер-загрози,  $a_{k_i}$  – значення, яке може приймати ознака кібер-загрози,  $i = 1, N_{A_k}$  – кількість значень що може прийняти ознака кібер-загрози.

Правило  $g_i$  приймає вигляд:

$$if (A_1 \wedge \dots \wedge A_n) \Rightarrow "name\_attack".$$

Розглянемо функціонування етапу виявлення кібер-загроз на основі еволюційних алгоритмів.

*Збір інформації, що вказує на можливу присутність кібер-загроз.* На даному етапі здійснюється моніторинг мережевого трафіку, відслідковування функціонування програмного забезпечення в операційній системі для виявлення кібер-загроз.

*Здійснення виявлення кібер-загроз на основі застосування генетичного алгоритму з подальшим віднесенням їх до певного класу.* Якщо правила однозначно виявили загрозу, крок 2 етапу виявлення загроз. Якщо виявлений підозрілий об'єкт (часткова відповідність ознак вище заданого порогу), крок 2 етапу навчання для класифікації та покращення механізму захисту.

*На основі одержаного результату здійснення відповідного реагування на можливу кібер-загрозу.* Отримання результатів щодо кібер-безпеки. Можливе блокування шкідливого чи небезпечного об'єкта кібер-загрози, повідомлення про безпеку, активація захисних механізмів.

**Експерименти.** Для оцінки ефективності запропонованого методу було проведено ряд експериментів. З цією метою було використано набір даних NSL-KDD [8]. Ця база даних містить стандартний набір даних, що підлягає перевірці, який включає в себе широкий спектр вторгнень.

Таким чином, для проведення експериментів набір даних NSL-KDD було використано з метою виділення множини ознак кібер-загроз, виділення з неї підмножини таких ознак і створення таких необхідних правил, які дозволять виявити кібер-загрози.

Оцінка ефективності запропонованого методу була апробована на таких атаках: ask, buffer\_overflow, ftp\_write, guess\_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, normal, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster.

В роботі подано результати проведених експериментів на прикладі nmap кібер-загрози, яка збирає інформацію про цільову систему перед початком атаки.

Перший етап експерименту полягав у використанні даних NSL-KDD для етапу навчання і генерації правил виявлення кібер-загроз. З цією метою було обрано 301 атаку типу «nmap». Другий етап експерименту полягав в безпосередньому виявленні вказаної атаки на основі згенерованих методом набору правил. Серед поданих нешкідливих з'єднань, були присутні різні типи загроз, серед них 1493 атаки типу «nmap».

На рис. 4 представлені приклади ознак кібер-загроз, які присутні в навчальній вибірці, на основі яких згенеровано правила виявлення кібер-загроз. Кожна ознака може прийняти будь-яке значення, що вказане для неї по горизонталі таблиці даних. Аналогічно були згенеровані ознаки нормальної поведінки з'єднання. На основі таких поведінок можливою є генерація правил, придатних для виявлення кібер-загроз.

duration	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
protocol_type	icmp	tcp	udp	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
service	eco	i	gopher	sql_net	private	netbios	ns	ldap	ecr	i	uucp	path	http	443	mtp	-	-	-	-	-
flag	SF	SH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
src_bytes	8	0	215	100	207	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
dst_bytes	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
land	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wrong_fragment	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
urgent	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hot	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
num_failed_logins	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
logged_in	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
num_compromised	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
root_shell	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
su_attempted	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
num_root	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
num_file_creations	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
num_shells	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
num_access_files	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
num_outbound_cmds	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
is_host_login	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
is_guest_login	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
count	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
srv_count	15	1	25	36	17	52	40	47	14	21	16	42	6	23	2	3	44	26	50	45
serror_rate	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
srv_serror_rate	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
rerror_rate	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
srv_rerror_rate	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
same_srv_rate	1	0,5	0,02	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
diff_srv_rate	0	1	0,04	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
srv_diff_host_rate	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
dst_host_count	2	255	4	1	3	99	35	185	155	25	5	39	223	220	37	52	139	27	143	73
dst_host_srv_count	46	1	112	169	80	255	83	75	77	61	158	153	190	223	194	54	131	24	10	215
dst_host_same_srv_rate	1	0	0,84	0,03	0,83	0,85	0,08	0,6	0,19	0,01	0,15	0,12	0,87	0,89	0,09	0,02	0,86	0,5	0,16	0,9
dst_host_diff_srv_rate	0	1	0,02	0,74	0,01	0,72	0,6	0,96	0,95	0,05	0,04	0,93	0,86	0,92	0,03	0,73	0,85	0,91	0,75	0,89
dst_host_same_src_port_rate	1	0,84	0,77	0,83	0,85	0,68	0,6	0,79	0,96	0,19	0,94	0,89	0,15	0,12	0,87	0,64	0,88	0,93	0,97	0,8
dst_host_srv_diff_host_rate	0,26	0	0,25	0,3	0,27	0,28	0,33	0,31	0,4	0,29	-	-	-	-	-	-	-	-	-	-
dst_host_serror_rate	0	1	0,77	0,68	0,79	0,96	0,94	0,89	0,64	0,88	0,93	0,97	0,8	0,91	0,95	0,71	0,2	0,56	0,75	0,86
dst_host_srv_serror_rate	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
dst_host_rerror_rate	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
dst_host_srv_rerror_rate	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**Рис. 4.** Ознаки загрози «nmap» з частиною значень, виявлених в ході кроку застосування генетичного алгоритму для мінімізації кількості ознак, необхідних для виявлення кібер-загроз та віднесення їх до певного класу

Для визначення ознак було встановлено наступні параметри генетичного алгоритму:

1. Мутація ( $p_m$ ) – ймовірність 0.05.
2. Покоління – 100.
3. Ймовірність кросинговеру – 1.
4. Особин в поколінні – 50.

Варто зауважити, що кожна особина в певному поколінні могла слугувати як множина ознак для розпізнавання загроз, якщо ні – відбирались найкращі нащадки попереднього покоління, відбувався процес кросинговеру і мутації. Суть застосування фітнес-функції полягала в зменшенні кількості необхідних ознак для виявлення загроз.

Експериментальні дослідження показали, що для виявлення атаки типу «nmap» необхідними є ознаки service, flag, src\_bytes, dst\_bytes, dst\_host\_srv\_diff\_host\_rate.

Отже, для виявлення загрози типу «nmap», було згенероване та використане правило:

If (service={eco\_i | gopher | sql\_net | private | netbios\_ns | ldap | ecr\_i | uucp\_path | http\_443 | mtp } ^ flag={SF | SH} ^ src\_bytes={8 | 0 | 215 | 100 | 207} ^ dst\_bytes={0} ^ dst\_host\_srv\_diff\_host\_rate={0,26 | 0 | 0,25 | 0,3 | 0,27 | 0,28 | 0,33 | 0,31 | 0,4 | 0,29}) => «nmap».

В наборі даних, використовуваних для навчання, за допомогою згенерованого правила було виявлено 301 з 301 атак типу nmap, проте також хибно зреагувало на 1 випадок нормального з'єднання.

На етапі виявлення згенероване правило дозволило виявити 1454 з 1501 атак типу «nmap», що становить 96.868%. Експериментальні дослідження також продемонстрували факт хибних спрацювань на рівні 7 випадків, що становить 0.466%.

Також експериментальні дослідження проводилися по відношенню інших типів кібер-загроз, результати яких представлені в таблиці 1.

Таблиця 1

#### Результати експериментальних досліджень для різних типів атак

Кібер-загроза	Рівень виявлення
nmap	96.868%
pod	100%
ipsweep	91.997%
smurf	92.252 %
teardrop	98.654%

Таким чином, запропонований метод продемонстрував можливість виявлення кібер-загрози з високою достовірністю.

**Висновки.** Запропоновано метод виявлення кібер-загроз на основі еволюційних алгоритмів. Метод дозволяє забезпечити реагування на нові кібер-загрози, забезпечуючи захист комп'ютерних систем від як відомих так і невідомих кібер-загроз. Робота системи виявлення нових загроз здійснюється на основі обробки зібраних в мережі та в комп'ютерній системі множини ознак кібер-загроз, виділення з неї підмножини таких ознак і створення таких необхідних правил, які дозволять виявити кібер-загрози.

Процес використовує генетичні алгоритми для мінімізації необхідних ознак виявлення кібер-загроз, що дозволяє знизити ресурсоемність процесу виявлення кібер-загроз.

#### References

1. Shapiro C. Information Rules: A Strategic Guide to the Network Economy / Carl Shapiro, Hal R. Varian, 1998, 352 p.
2. Pels M. Host-Based Intrusion Detection Systems Faculty of Science, Informatics Institute, University of Amsterdam / D. B. P., M. Pels [Technical Report], 2005.
3. Hassan M. Hybrid Intelligent System / Mohamed Hassan, 2013.
4. Dhammi A. Behavior Analysis of Malware Using Machine Learning / Arshi Dhammi.
5. Pancenko T.V. Genetic algorithms / T.V. Pancenko, 2007, 87 p.
6. Gowher M.P. Genetic algorithms in intrusion detection systems: A survey / M.P. Gowher, S. Kumar, 2014.
7. Intrusion detector learning. URL: <http://kdd.ics.uci.edu/databases/kddcup99/task.html>.
8. NSL-KDD dataset. URL: <http://www.unb.ca/cic/datasets/nsl.html>.

Рецензія/Peer review : 06.11.2017 р.

Надрукована/Printed :06.12.2017 р.  
Рецензент: д.т.н., проф. Боровик О.В.