МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРНІГІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

ПЕТРЕНКО ТАРАС АНАТОЛІЙОВИЧ

УДК 004.056.53

ДИСЕРТАЦІЯ
МЕТОДИ ТА МОДЕЛІ ЕКСПЕРТНИХ СИСТЕМ РОЗПІЗНАВАННЯ КІБЕРАТАК НА ОСНОВІ КЛАСТЕРИЗАЦІЇ РЕАЛІЗАЦІЙ ОЗНАК

Спеціальність: 05.13.21 – системи захисту інформації
Галузь знань: 12 – Інформаційні технології

Подається на здобуття наукового ступеня кандидата технічних наук (доктора філософії)

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Т.А.Петренко

Науковий керівник *Лахно Валерій Анатолійович, доктор технічних наук, професор*

Київ – 2019

АНОТАЦІЯ

Петренко Т.А. Методи та моделі експертних систем розпізнавання кібератак на основі кластеризації реалізацій ознак. – Кваліфікаційна наукова праця на правах рукопису.

Дисертації на здобуття наукового ступеня кандидата технічних наук (доктора філософії) за спеціальністю 05.13.21 – системи захисту інформації (125 – Кібербезпека*). Національний авіаційний університет Міністерства освіти і науки України, Київ, 2019.*

Дисертаційна робота містить результати досліджень, які спрямовані на подальший розвиток методів та моделей для адаптивних систем розпізнавання кібератак на основі кластеризації реалізацій ознак. Запропоновано структурну схему здатної до самонавчання експертної системи (ЕС) з інформаційної безпеки.

На основі проведеного аналізу доступних наукових публікацій з'ясовано, що складність застосування до інтелектуальних систем розпізнавання цільових кібератак формалізованого апарату аналізу й синтезу, полягає в тому, що конкретний інформаційний комплекс та їх підсистеми інформаційної безпеки (ІБ) складаються з різнорідних елементів, які описуються із використанням різних моделей. Показано, що застосування елементів адаптивного захисту інформації може бути засноване на використанні новітніх методів інтелектуального розпізнавання кібератак.

Вперше розроблено модель експертної системи у складі інтелектуальних систем виявлення вторгнень, у якій, на відміну від існуючих, застосовується процедура нечіткої кластеризації реалізацій ознак кібератак та наступна корекція вирішальних правил, що дозволяє створювати адаптивні механізми самонавчання систем інтелектуального розпізнавання кібератак.

Вперше запропоновано застосовувати в якості оціночного показника ефективності навчання експертної системи модифіковану інформаційну умову функціональної результативності, яка ґрунтується на ентропійному та

інформаційно-дистанційному критерії Кульбака – Лейблера, та, на відміну від існуючих, дозволяє отримувати вхідну навчальну матрицю, яка використовується як об’єкт навчання, й будувати коректні вирішальні правила розпізнавання кібератак на критично важливі інформаційні системи.

Удосконалено метод розбиття простору реалізацій ознак на кластери в ході реалізації процедури розпізнавання кібератак, який відрізняється від існуючих, одночасною оптимізацією при обчисленні контрольних допусків у ході аналізу експертною системою важко пояснюваних реалізацій ознак об’єктів спостереження, та дозволяє на кожному кроці навчання змінювати перевірочні допустимі відхилення для всіх реалізацій ознак кібератак одночасно.

Удосконалено метод навчання експертної системи, який являє собою ітераційну процедуру пошуку глобального максимуму інформаційної умови функціональної результативності, та, на відміну від існуючих, дозволяє попереджати можливі випадки поглинання одним класом об’єктів розпізнавання базових реалізацій ознак іншого класу, враховує відомі статистичні параметри кластеризації реалізацій ознак об’єктів спостереження, а також помилки під час завдання на прийняття рішення в ході процедур машинного навчання.

Набули подальшого розвитку імітаційні моделі для композитної побудови систем інтелектуального виявлення кібератак за рахунок одночасної оптимізації контрольних допусків в ході аналізу об’єктів розпізнавання, що дозволяє проводити їх дослідження, здійснювати вибір раціональних способів протидії і нейтралізації наслідків, аналізувати більш складні і раніше невідомі види кібератак на критично важливі інформаційні системи.

Практичне значення одержаних результатів полягає в тому, що дисертантом розроблена програмна реалізація ЕС для розпізнавання кібератак, що дозволяє підвищити ефективність розпізнавання, в залежності від класу кібератак, до 70 % – 99%, та на 15–20% зменшити час

налагодження проектів систем захисту інформації КВІС за рахунок імітаційного моделювання кібератак. Здатна до самонавчання ЕС «Analyzer of cyberthreats» («Аналізатор загроз»), дозволяє враховувати відомі статистичні параметри кластеризації реалізацій ознак об'єктів розпізнавання, а також помилки під час завдання на прийняття рішення під час машинного навчання. Запропонована ЕС дозволяє підвищити результативність функціонування систем кібербезпеки.

Результати роботи впроваджено в науково-технічних розробках ТОВ «Захист інформації» та у навчальний процес у Чернігівському національному технологічному університеті, а також у Національному авіаційному університеті.

Ключові слова: захист інформації, інформаційна безпека, інтелектуальне розпізнавання кібератак, кластеризація ознак.

ABSTRACT

Petrenko T.A. Methods and models of expert systems of recognition of cyber attacks on the basis of clustering of features. - Manuscript.

Thesis for a candidate degree in technical sciences by speciality 05.13.21. - information security systems. National Aviation University of the Ministry of Education and Science of Ukraine, Kyiv, 2019.

The dissertation contains the results of researches aimed at further development of methods and models for adaptive systems of recognition of cyber attacks on the basis of clusterization of the implementation of features. A structural scheme of an expert system for information security capable of self-education is proposed.

On the basis of the analysis of available scientific publications, it was found that the complexity of application to intelligent recognition systems of target cyber attacks of the formalized apparatus of analysis and synthesis is that a specific information complex and their subsystems of information security consist of heterogeneous elements that are described using different models. It is shown that

the application of elements of adaptive information protection can be based on the use of advanced methods of intelligent recognition of cyber attacks.

For the first time, an expert system model has been developed as part of intelligent intrusion detection systems, in which, unlike existing ones, the procedure of fuzzy clusterization of the implementation of cyber attacks and the subsequent correction of decisive rules is applied, which allows for the creation of adaptive self-learning mechanisms for intelligent cyber attacks detection systems.

For the first time, it is suggested to use the modified informational condition of functional effectiveness as an estimator of the effectiveness of the training of the expert system based on the entropy and informational-remote criterion of the Kulbak-Leibler, and, unlike the existing one, allows to receive the entrance educational matrix, which is used as an object of study , and build correct decisive rules for recognizing cyber attacks on critical information systems.

The method of partitioning the realization of features into clusters in the course of implementing the cyber attacks recognition procedure, which differs from existing, is simultaneously optimized when calculating control tolerances during the analysis by an expert system of difficult explanatory implementations of the objects of observation, and allows for each step of the training to change the test tolerance deviations for all implementations of cyber attacks simultaneously.

The method of teaching the expert system is improved, which is an iterative procedure for finding the global maximum of the information condition of functional efficiency, and, unlike the existing one, prevents possible cases of one object acquisition of objects of recognition of basic realizations of signs of observation objects, as well as errors during the task of making decisions in the course of machine learning procedures.

Further development of simulation models for the composite construction of intelligent detection systems for cyber attacks by simultaneously optimizing control tolerances during the analysis of recognition objects, allowing them to conduct research, to select rational methods of counteraction and neutralization of

consequences, to analyze more complex and previously unknown types of cyber attacks on critical information systems.

The practical significance of the results is that the dissertation developed a software implementation of the expert system for the recognition of cyber attacks, which allows to increase the efficiency of recognition, depending on the class of cyber attacks, up to 70-99%, and reduce the time of debugging of the information security systems projects by 15-20% CYIS at the expense of simulation of a cyber attacks. An EU-based self-learning tool, "Analyzer of Cyberthreats", allows you to take into account the known statistical parameters of clustering implementations of features of recognition objects, as well as errors during the decision making task during machine learning. The proposed expert system allows to increase the efficiency of functioning of cybersecurity systems.

The results of the work have been implemented in the scientific and technical developments of "Protection of Information" Ltd. and in the educational process at the Chernihiv National Technological University, as well as at the National Aviation University.

Key words: protection of information, information security, intellectual recognition of cyber attacks, clustering of the implementation of features.

Список публікацій здобувача

Skopus

1. V. Lakhno, Y. Tkach, T. Petrenko, S. Zaitsev and V. Bazylevych, “Development of adaptive expert system of information security using a procedure of clustering the attributes of anomalies and cyber attacks”, *Eastern-European Journal of Enterprise Technologies,* no. 6/9 (84), pp. 32–44, 2016.

2. V. Lakhno, S. Zaitsev, Y. Tkach, T. Petrenko, “Adaptive expert systems development for cyber attacks recognition in information educational systems on the basis of signs’ clustering”, *Advances in Intelligent Systems and Computing*, v. 754. pp. 673–682, 2018.

Праці у закордонних виданнях:

3. T. Petrenko, “Design of adaptive system for detection of cyber-attacks”, *MEST Journal: Management, Education, Science & Society, Technologies,* v. 5, pp. 78–85, 2017.

Праці у фахових виданнях:

4. В.А. Лахно, Т.А Петренко і М.В. Пирог, “Моделювання роботи адаптивної системи розпізнавання кібератак в умовах неоднорідних потоків запитів в модулях e-business”, *Безпека інформації*, т. 22, № 2, с. 135–142, 2016.

5. В.А. Лахно, А.М. Терещук и Т.А. Петренко, “Совершенствование киберзащиты информационных систем за счет адаптивных технологий распознавания кибератак”, *Захист інформації,* т.18, № 2, с. 99–106, 2016.

6. Д.Б. Мехед, Ю.М. Ткач Ю.М., В.М. Базилевич і Т.А. Петренко, “Аналіз загроз інформаційної безпеки в мережах стандарту IEEE 802.11”, *Захист інформації*, т. 17, №4. с. 285–291, 2015.

Праці які засвідчують апробацію матеріалів дисертації:

7. Т.А. Петренко, “Прикладні аспекти захисту конфіденційної інформації на підприємстві”, *Новітні технології у науковій діяльності і навчальному процесі: матеріали Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених (Чернігів, 23-24 квітня 2014 р.): тези доповідей,* Ч., 2014, С. 26–28.

8. Ю.В. Козинець та Т.А. Петренко “Дослідження процесу управління інформаційною безпекою”, *Новітні технології у науковій діяльності і навчальному процесі: матеріали Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених (Чернігів,28 квітня 2015 р.): тези доповідей*, Ч., 2015, С. 26–27.

9. Д.Б. Мехед, В.М. Базилевич і Т.А. Петренко, “Оцінка ризиків інформаційної безпеки на підприємстві”, *Стан та удосконалення безпеки інформаційно-комунікаційних систем “SITS-2015”: матеріали VII Всеукраїнської науково-практичної конференції (Миколаїв-Коблево, 9-12 червня 2015 р.): збірник тез доповідей*, Миколаїв-Коблево, 2015, С. 59 – 61.

10. Т.А. Петренко, В.А. Лахно, Г.С. Григорян, “Розробка адаптивної системи розпізнавання кіберзагроз”, *Безпека українського суспільства в концепції вступу в постіндустріальне суспільство ЄС: Наукові доповіді та тези учасників науково-практичної конференції (м. Київ, 16 грудня 2015 р.),* К., 2015. С. 66–76.

11. Ю.М. Ткач, Д.Б. Мехед, В.М. Базилевич і Т.А. Петренко, “Аналіз загроз інформаційної безпеки в WI-FI мережах”, *Актуальні питання забезпечення кібербезпеки та захисту інформації: тези доповідей учасників II Міжнародної науково-практичної конференції (Закарпатська область, Міжгірський район, село Вернє Студене, туристичний комплекс «Едельвейс», 24-27 лютого 2016 року)*, К., 2016, С. 151–155.

12. В.А. Лахно, А.М. Терещук и Т.А. Петренко “Адаптивные системы распознавания кибератак на критически важные компьютерные системы”, *II міжнародна науково-практична конференція “Актуальні проблеми*

моделювання ризиків і загроз виникнення надзвичайних ситуацій на об'єктах критичної інфраструктури": Наукові доповіді та тези учасників науково-практичної конференції (м. Київ, 26-28 травня 2016р.), К., 2016, С. 136-141.

13. T. Petrenko and V. Lakhno, "A model developed for teaching an adaptive system of recognising cyberattacks in information systems", *The seventh world congress "Aviation in the XXI-st century" Safety in Aviation and Space Technologies, Kyiv, NAU, September 19-21*, 2016, pp. 406-408.

14. В.А. Лахно и Т.А. Петренко, "Система интеллектуальной поддержки принятия решений в слабо формализуемых задачах обеспечения кибербезопасности", *Комплексне забезпечення якості технологічних процесів та систем (КЗЯТПС – 2017): матеріали тез доповідей VII міжнародної науково-практичної конференції (м. Чернігів, 24–27 квітня 2017 р.)*, Ч., 2017, Т. 2, С. 107-108.

15. Т.А. Петренко та В.А. Лахно, "Метод та моделі адаптивних експертних систем розпізнавання кібератак на основі кластеризації ознак", *ITSEC: Безпека інформаційних технологій: VII міжнародна науково-технічна конференція, 16-18 травня 2017 р.*, м. Київ, НАУ, 2017, С. 77-78.

Праці в інших виданнях за темою дисертації:

16. Т.А. Петренко, "Інформаційна безпека в сучасних умовах", *Вісник Чернігівського державного інституту права, соціальних технологій та праці,* №2, с. 98–102, 2009.

ЗМІСТ

СПИСОК УМОВНИХ СКОРОЧЕНЬ……. 12

ВСТУП…….. 13

РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ ТА МОДЕЛЕЙ РОЗПІЗНАВАННЯ КІБЕРАТАК НА КРИТИЧНО ВАЖЛИВІ ІНФОРМАЦІЙНІ СИСТЕМИ……..…….. 20

1.1. Проблеми виявлення та розпізнавання складних атак в умовах зростання деструктивних впливів на кібербезпеку критично важливих інформаційних систем…….. 20

1.2. Огляд та аналіз методів які використовуються для побудови систем інтелектуального розпізнавання кібератак…….. 33

1.3. Висновки до першого розділу……... 46

РОЗДІЛ 2. МОДЕЛЬ АДАПТИВНОЇ СИСТЕМИ ІНТЕЛЕКТУАЛЬНОГО РОЗПІЗНАВАННЯ КІБЕРАТАК ІЗ ВИКОРИСТАННЯМ ПРОЦЕДУРИ НЕЧІТКОЇ КЛАСТЕРИЗАЦІЇ РЕАЛІЗАЦІЙ ОЗНАК…….……. 49

2.1. Формалізована постановка задачі інформаційного синтезу адаптивної системи розпізнавання кібератак……... 50

2.2. Інформаційні критерії функціональної ефективності навчання адаптивної системи розпізнавання…… 75

2.3. Висновки до другого розділу……. 99

РОЗДІЛ 3. МЕТОДИ КЛАСТЕРИЗАЦІЇ ТА ОПТИМІЗАЦІЇ КОНТРОЛЬНИХ ДОПУСКІВ НА РЕАЛІЗАЦІЇ ОЗНАКИ РОЗПІЗНАВАННЯ КІБЕРАТАК…….. 101

3.1. Метод кластеризації реалізацій ознак при виявлення складних кібератак…….. 101

3.2. Імітаційне моделювання адаптивної системи інтелектуального розпізнавання із використанням процедури нечіткої кластеризації та паралельної оптимізації контрольних відхилень для реалізацій ознак кібератак ……..……..... 123
3.3. Висновки до третього розділу…….. 137
РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ АДАПТИВНОЇ ЕКСПЕРТНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ КІБЕРАТАК …..…….. 139
4.1. Методологія проведення експериментального дослідження адаптивної експертної системи розпізнавання кібератак…….. 139
4.2. Результати експериментальних досліджень…… 159
4.3. Висновки до четвертого розділу…….. 175
ВИСНОВКИ ……. 177
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ……... 180
ДОДАТОК А. Відомості щодо впровадження результатів роботи…… 200
ДОДАТОК Б. Лістинг ЕС «Analyzer of cyberthreats»..... 203

СПИСОК УМОВНИХ СКОРОЧЕНЬ

АЕС – адаптивна експертна система

АС – автоматизована система

АСК – автоматизована система керування

АСР – адаптивна система розпізнавання

АСРКА – адаптивна системи розпізнавання кібератак

БВПНМ – безпомилкові вирішальні правила за відповідною навчальною матрицею реалізацій ознак

ЕС – експертна система

ІБ – інформаційна безпека

ІЕТН – інформаційно екстремальна технологія навчання

ІС – інформаційна система

ІТС – інтелектуальні системи та технології

КВІС – критично важлива інформаційна система

КВКС – критично важлива комп’ютерна система

КМС – комп'ютерні мережі та системи

НСД – несанкціонований доступ

ОР – об’єкт розпізнавання

ОС – операційна система

ППЗ – прикладне програмне забезпечення

РБНМ – радіально-базисні нейронні мережі

СВВ – система виявлення вторгнень

СІРКЗ – система інтелектуального розпізнавання кіберзагроз

СКБД – система керування базами даних

СППР – система підтримки прийняття рішень

СРВ – система розпізнавання вторгнень

СРКА – система розпізнавання кібератак

ШНМ – штучна нейронна мережа

ВСТУП

Актуальність теми. Останні десятиліття характеризуються швидким зростанням і розвитком комп'ютерних мереж та систем, з метою забезпечення кібербезпеки яких розроблено безліч систем, що використовують різні техніки опрацювання даних для виявлення нелегітимної діяльності. Більшість класичних систем виявлення кібератак характеризуються рядом недоліків (недостатня масштабованість, відсутність гнучкості тощо), що накладає додаткові обмеження на області їх застосування. Як показує досвід останніх років, кіберзлочинці все частіше використовують унікальні, ще не відомі IT-індустрії шкідливі програми, уразливості і способи кібератак.

Активне розширення інформаційно-комунікаційного середовища (ІКС) та критично-важливих інформаційних систем (КВІС) у багатьох державах світу, супроводжується виникненням нових загроз для кібербезпеки (КБ), про що свідчить зростання кількості інцидентів, пов’язаних із захистом інформації, а також виявлених уразливостей у КВІС. Зростання інтересу до проблематики КБ та інформаційної безпеки (ІБ), викликало за останнє десятиліття сплеск досліджень в області розроблення ефективних систем виявлення й запобігання кіберзагрозам.

Проблеми, пов'язані із захистом інформації й забезпеченням ІБ та кібербезпеки, успішно вирішують провідні вітчизняні й закордонні вчені: Блінцов В.С., Гайкович В.Ю., Герасименко В.А., Глова В.І., Грищук Р. В., Дудикевич В.Б., Клейменов С.А., Котенко В.І., Ахмад Д.М., Кеммерер Р., Віджна Дж., Касперський К., Козіол Дж., Коен Ф., Мун С., Норткат С., Стенг Д., Форрестол Д., Ховард М., Atighetchi M., Campbell R.H., Chapman C., Chi S.-D., Dawkins J., Endler M., Hariri S., Jha S., Kaiser G., Keromytis A.D., Misra V., Nieh J., Parekh J., Rubensteiny D., Silva F., Stolfo S. та ін.

Протистояти постійному зростанню кількості й складності деструктивних впливів на КВІС можна, зокрема, й використовуючи

адаптивні інтелектуальні системи розпізнавання кіберзагроз (СІРКЗ). Термін «адаптація» для СІРКЗ можна тлумачити як процес цілеспрямованої зміни структури, алгоритму або параметрів системи з метою підвищення ефективності її функціонування. Зокрема, при цьому можна використовувати методи інтелектуальних технологій машинного навчання (МІТН). У свою чергу, МІТН можуть використовувати моделі, які, ґрунтуються на максимізації інформаційної спроможності СІРКЗ шляхом імплементації в їхній склад експертних систем або систем підтримки прийняття рішень та використовуванні в процесі навчання додаткових інформаційних обмежень, які, наприклад, стосуються реалізацій ознак кібератак в межах відомих та нових класів вторгнень.

Отже, дослідження спрямовані на подальший розвиток моделей та методів кіберзахисту на основі застосування адаптивних, здатних до самонавчання систем інтелектуального розпізнавання кібератак є актуальними.

Зв'язок роботи з науковими програмами, планами, темами. Розробка основних положень дослідження здійснювалася відповідно до планів НДР і договорів, що виконуються протягом 2015–2018 р.р. у Чернігівському національному технологічному університеті, Приватному вищому навчальному закладі “Європейський університет”: 1) НДР “Методи та засоби забезпечення безпеки ресурсів інформаційних систем” (№ держ. реєстрації 0117U003187); 2) НДР “Інформаційне та програмне забезпечення, математичне моделювання складних систем” (№ держ. реєстрації 0114U005430). При виконанні робіт дисертант брав участь у розробці методів забезпечення ІБ і математичних моделей СЗІ, як виконавець окремих розділів.

Мета дослідження – підвищення ефективності систем інтелектуального розпізнавання кібератак для КВІС на основі створення здатної до самонавчання експертної системи (ЕС), яка враховує відомі статистичні параметри кластеризації реалізацій ознак кібератак, як об’єкта

спостереження.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

1) проаналізувати методи які використовуються для побудови систем інтелектуального розпізнавання кібератак в умовах зростання кількості та складності цільових атак, які характеризуються реалізаціями ознак, які важко пояснити, а також ураховують відомі статистичні параметри кластеризації реалізацій ознак кібератак;

2) розробити модель експертної системи та метод її навчання із використанням процедури нечіткої кластеризації реалізацій ознак кібератак та можливістю корекції вирішальних правил, що дозволить створювати адаптивні механізми самонавчання систем кіберзахисту;

3) удосконалити методи навчання експертної системи та визначити раціональну кількість кластерів у просторі реалізацій ознак кібератак, що дозволить зменшити час навчання;

4) розробити експертну систему розпізнавання кібератак (об’єктів розпізнавання – ОР), яка у взаємодії із іншими системами кіберзахисту дозволить підвищити рівень розпізнавання складних цільових кібератак;

5) провести імітаційні дослідження та натурну апробацію моделі експертної системи.

Об’єкт дослідження – процеси розпізнавання кібератак у критично важливих інформаційних системах.

Предмет дослідження – методи та моделі здатних до самонавчання адаптивних експертних систем у складі комплексів виявлення вторгнень у критично важливих інформаційних системах.

Методи дослідження. У процесі дослідження, враховуючи особливості предметної галузі та сформульованих задач, використано: методи теорії захисту інформації та кластерного аналізу (для побудови алгоритму розбиття простору реалізацій ознак на кластери); методи прикладної статистики, оптимізації та комп'ютерного моделювання (для імітаційного моделювання); моделі інтелектуальних технологій машинного навчання (МНав) та теорію

нечітких множин (для побудови моделі оціночного показника ефективності навчання та процедури нечіткої кластеризації реалізацій ознак розпізнавання у ЕС), принципи й методи об'єктно-орієнтованого програмування для створення ЕС у складі СІРКЗ; методологію тесту на проникнення (для експериментальної перевірки розроблених моделей).

Наукова новизна одержаних результатів.

У межах виконаних досліджень отримані такі наукові результати:

1. Уперше:

– розроблено модель експертної системи у складі інтелектуальних систем виявлення вторгнень, у якій, на відміну від існуючих, застосовується процедура нечіткої кластеризації реалізацій ознак кібератак та наступна корекція вирішальних правил, що дозволяє створювати адаптивні механізми самонавчання систем інтелектуального розпізнавання кібератак;

– запропоновано застосовувати в якості оціночного показника ефективності навчання експертної системи модифіковану інформаційну умову функціональної результативності, яка ґрунтується на ентропійному та інформаційно-дистанційному критерії Кульбака – Лейблера, та, на відміну від існуючих, дозволяє отримувати вхідну навчальну матрицю, яка використовується як об’єкт навчання, й будувати коректні вирішальні правила розпізнавання кібератак на критично важливі інформаційні системи.

2. Удосконалено:

– метод розбиття простору реалізацій ознак на кластери в ході реалізації процедури розпізнавання кібератак, який відрізняється від існуючих, одночасною оптимізацією при обчисленні контрольних допусків у ході аналізу експертною системою складних реалізацій ознак об’єктів спостереження, та дозволяє на кожному кроці навчання змінювати перевірочні допустимі відхилення для всіх реалізацій ознак кібератак одночасно;

– метод навчання експертної системи, який являє собою ітераційну процедуру пошуку глобального максимуму інформаційної умови

функціональної результативності, та, на відміну від існуючих, дозволяє попереджати можливі випадки поглинання одним класом об’єктів розпізнавання базових реалізацій ознак іншого класу, враховує відомі статистичні параметри кластеризації реалізацій ознак об’єктів спостереження, а також помилки під час завдання на прийняття рішення в ході процедур машинного навчання.

3. Набули подальшого розвитку:

– імітаційні моделі для композитної побудови систем інтелектуального виявлення кібератак за рахунок одночасної оптимізації контрольних допусків в ході аналізу об’єктів розпізнавання, що дозволяє проводити їх дослідження, здійснювати вибір раціональних способів протидії і нейтралізації наслідків, аналізувати більш складні і раніше невідомі види кібератак на критично важливі інформаційні системи.

Практична значимість наукових результатів.

1. Запропоновані в дисертації моделі та методи, у рамках розробленої ЕС з розпізнавання кібератак, доведені до практичної реалізації шляхом імплементації програмних модулів в MatLAB та Delphi, що дозволяє підвищити ефективність розпізнавання, в залежності від класу кібератак, до 70 % – 99%, та на 15–20% зменшити час налагодження проектів СЗІ КВІС (КВКС) за рахунок імітаційного моделювання кібератак.

2. Запропоновані в дисертації модель та метод, доведені до практичної реалізації шляхом створення відповідних програмних модулів здатної до самонавчання експертної системи «Analyzer of cyberthreats» («Аналізатор загроз»), що дозволяє враховувати відомі статистичні параметри кластеризації реалізацій ознак об’єктів розпізнавання, а також помилки під час завдання на прийняття рішення під час машинного навчання. Запропонована ЕС дозволяє підвищити результативність СІРКЗ, які працюють у складі СЗІ КВІС (КВКС) до 98 %. ЕС «Analyzer of cyberthreats» впроваджена в ТОВ «Захист інформаії» (акт №134 від 16.11.2018р.).

3. Запропоновані програмні продукти можуть бути використані для забезпечення кіберзахисту державних КВІС, модернізації існуючих СІРКЗ.

4. Результати досліджень впроваджено в навчальний процес у Чернігівському національному технологічному університеті (акт №9 від 5.09.2018 р.) та Національному авіаційному університеті (акт від 3.09.2018 р.).

Достовірність і обґрунтованість наукових результатів підтверджується: математичною адекватністю розробленої моделі ЕС; теоретичною та практичною верифікацією методів навчання ЕС; практичним впровадженням ЕС для інтелектуального розпізнавання кібератак на КВІС; збігом результатів імітаційного моделювання процедури розпізнавання кібератак і тестів на проникнення у КВІС.

Особистий внесок здобувача.

Усі основні результати одержані здобувачем особисто. У роботах, опублікованих із співавторами, здобувачу належать: [6, 11] – аналіз загроз інформаційної безпеки бездротових мереж, [4, 10, 12, 13] – категорійна модель адаптивної системи розпізнавання кібератак; [5, 15] – модель та метод навчання адаптивної системи розпізнавання кібератак; [1] – структурна схема адаптивної експертної системи, категорійна модель для визначення інформаційного критерію функціональної результативності навчання експертної системи; [2, 14] – модель адаптивної системи розпізнавання, модифікована інформаційна умова функціональної результативності навчання експертної системи; [8, 9] – складові процесу управління інформаційною безпекою.

Апробація результатів дисертації. Основні результати роботи були представлені та обговорені на наступних конференціях: 1) Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Новітні технології у науковій діяльності і навчальному процесі», Чернігів, 2014 р., 2015 р. 2) Круглому столі “Безпека українського суспільства в концепції вступу в постіндустріальне суспільство ЄС”, Київ, 2015 р.; 3) VII

Всеукраїнській науково-практичній конференції «Стан та удосконалення безпеки інформаційно-комунікаційних систем «SITS-2015»; 4) II Міжнародній науково-практичній конференції (МНПК) «Актуальні питання забезпечення кібербезпеки та захисту інформації», Закарпатська область, Міжгірський район, с. Вернє Студене, 24–27 лютого 2016 р. 5) VII Всесвітньому конгресі "Авіація у XXI столітті" – "Безпека в авіації та космічні технології", Київ, 2016р.; 6) VII МНПК Комплексне забезпечення якості технологічних процесів та систем (КЗЯТПС – 2017), Чернігів, 24–27 квіт. 2017 р. 7) II МНПК «Актуальні проблеми моделювання ризиків і загроз виникнення надзвичайних ситуацій на об'єктах критичної інфраструктури», Київ, 26–28 травня 2016 р., 8) XXIII МНПК «Інформаційні технології в економіці, менеджменті і бізнесі. Проблеми науки, практики і освіти», Київ, 29 листопада 2017 р.

Публікації. За результатами дослідження опубліковано 16 наукових праць в яких повністю відображені основні результати дисертації (з них 3 одноосібно). В їх число входять 2 статті що входять до наукометричної бази Scopus [1, 2], 1 стаття у наукових періодичних виданнях інших держав[3], 3 статті у фахових виданнях України [4, 5, 6] а також 10 матеріалів і тез доповідей на всеукраїнських та міжнародних конференціях [7-16].

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ ТА МОДЕЛЕЙ РОЗПІЗНАВАННЯ КІБЕРАТАК НА КРИТИЧНО ВАЖЛИВІ ІНФОРМАЦІЙНІ СИСТЕМИ

1.1. Проблеми виявлення та розпізнавання складних атак в умовах зростання деструктивних впливів на кібербезпеку критично важливих інформаційних систем

Останні десятиліття характеризуються швидким зростанням і розвитком комп'ютерних мереж та систем (КМС), з метою забезпечення кібербезпеки яких розроблено безліч методів, що використовують різні техніки обробки даних для виявлення нелегітимної діяльності. Наскільки ефективно будуть функціонувати сучасні системи розпізнавання кібератак (СРКА) багато в чому залежить від технологій які використовуються на етапах збору та опрацювання моніторингової інформації яка фіксує спроби втручання у роботу КМС. СРКА повинні, в першу чергу, виявляти активність кіберзлочинців на попередніх стадіях реалізації кібератак на інформаційні ресурси, зокрема й критично важливі [1, 38, 39,43, 46, 166], а по друге, мати здатність постійно поповнювати репозиторій шаблонів кібератак по мірі зміни атакуючими стратегій та тактики вторгнень [1, 46, 133]. Як показує досвід створення ефективних СРКА [133, 146, 173], на сьогодні найбільш результативним є підхід, який передбачає комбінацію інноваційних моніторингових систем за нелегітимною діяльністю кіберзлочинців на початкових стадіях їх діяльності, та систем інтелектуального розпізнавання кіберзагроз (СІРКЗ) або кібератак, направлених на конкретні уразливості критично важливих інформаційних систем (КВІС). Реалізація подібної концепції дозволить розв’язувати нетривіальні задачі в межах управління процесами кіберзахисту розподілених КВІС як комплекс взаємопов’язаних завдань, скоординованих між собою.

Більшість критично важливих комп'ютерних систем (КВКС), інформаційних систем (ІС) або автоматизованих систем керування (АСК) мають архітектуру яка включає чотири ієрархічних рівня: 1) прикладне програмне забезпечення (ППЗ); 3) системи керування базами даних (СКБД); 3) операційні системи (ОС); 4) мережу, яка забезпечує взаємодію вузлів КВІС, використовуючи, зокрема, протоколи TCP/IP, IPS/SPX, SMB/NetBIOS та ін. [3, 4].

За даними багатьох досліджень [2, 6, 7, 8, 9], до найбільш поширених типів кібератак можна віднести: несанкціонований доступ (НСД) к паролям користувачів КВІС [6, 17]; віддалене несанкціоноване виконання команд в системі (наприклад, внаслідок помилки «переповнення буфера») [13, 44]; нелегітимне отримання прав доступу в систему [7, 8]; атаки типу «відмова в обслуговуванні» [76, 202, 208]; використовування шкідливого програмного забезпечення (ПЗ) [2, 3, 4, 29, 30]. Труднощі ефективного динамічного формування параметрів оцінки кіберзагроз для КВІС полягають у тому, що: 1) розмір зони пошуку визначається потужністю початкових множин реалізацій ознак атак – об'єктів розпізнавання (ОР); 2) значна кількість факторів різної природи, які впливають на стан кібербезпеки КВІС, додають ймовірнісні складові до завдань СРКА [12, 14, 32, 40, 66, 72].

Оцінка загроз ІБ КВІС та КВКС включає дві складові: ситуаційний аналіз й безпосередню процедуру розпізнавання [47, 62, 68, 69, 77]. Ситуаційний аналіз дозволяє дослідити параметри функціонування апаратно-програмного забезпечення КВІС, при цьому доцільно згрупувати однотипні дані й оцінити їх окремо по кожній групі [11, 45, 52, 53, 83]. Результати такого аналізу наведені на рис. 1.1 та 1.2. Розпізнавання кібератак передбачає комплексний деталізований аналіз факторів, які впливають або можуть вплинути на ІБ та кібербезпеку КВІС. ОР, у відповідності до класів KL [24, 34, 41, 81], можна поділити на три базові групи [86, 105, 112]: «Потенційні» [89, 91]; «Реальні» [96, 101]; «Спрямовані» [76, 90, 92].

Крім того, як показали дослідження [131, 138, 143, 174, 206], вимоги до рівня складності «успішних кібератак» проти КВІС у промисловості, на транспорті та зв'язку (після того як зловмисник отримав доступ до цілі кібератаки), знизилася з максимального рівня – більш ніж на 90 % в 2004 році, до 48 % в 2015 році.

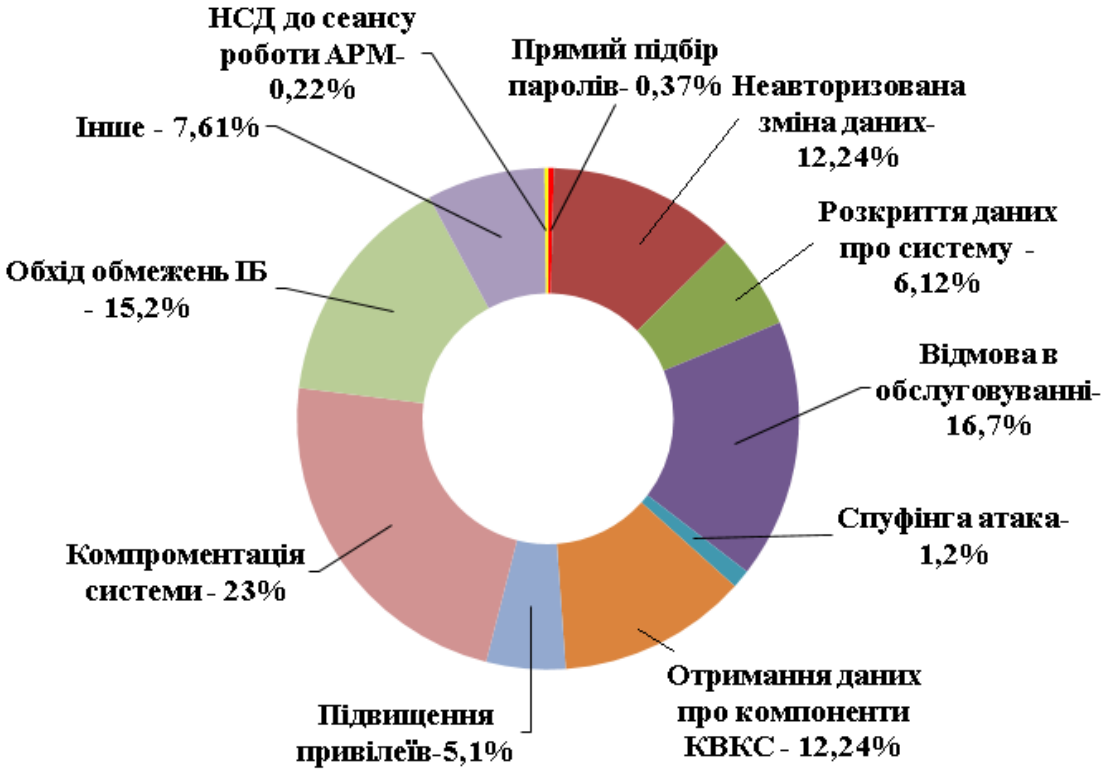


Рис. 1.1. Розподіл частки найбільших кіберзагроз для КВІС, КВКС та АС підприємств
(Джерела: [131, 138, 143, 174, 181, 186, 187, 206])

Як свідчить статистичний аналіз, в той же період відсоток уразливостей середньої складності збільшився з 5 % до 47 %, рис. 1.3. Розкриття інформації якій притаманні складні уразливості, протягом останніх десятиліть, не змінює свою частку, залишаючись в середньому на рівні 4 % – 4,5 % [143, 174, 181, 186, 187].

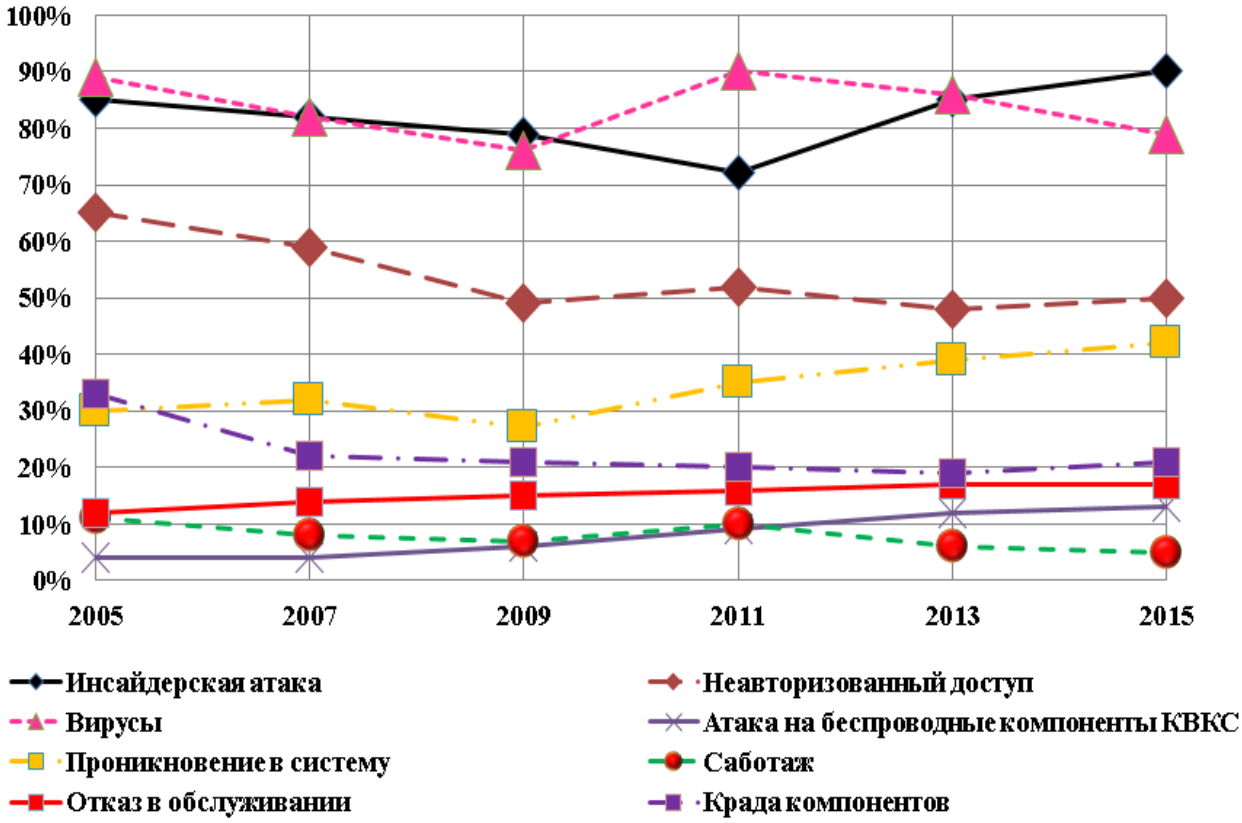


Рис. 1.2. Динаміка найбільших кіберзагроз
для КВІС, КВКС та АС підприємств
(Джерела: [93, 131, 138, 143, 174, 181, 186, 187, 206])

У зловмисників є кілька точок входу в комп'ютерні системи, що дозволяє в майбутньому скомпрометувати АСК або ІС. КВІС можна заразити різними способами. Наприклад, віруси (експлойти) можуть бути завантажені через USB-з'єднання або через мережеві інтерфейси. Як правило, кількість виявлених уразливостей корелює з кількістю опублікованих експлойтів. Так з лютого 2011 р. по вересень 2015 р. було опубліковано понад 150 експлойтів [143, 174, 181, 186, 187], тобто, це в вісім разів більше, ніж за період з 2005 р. по 2010 р.

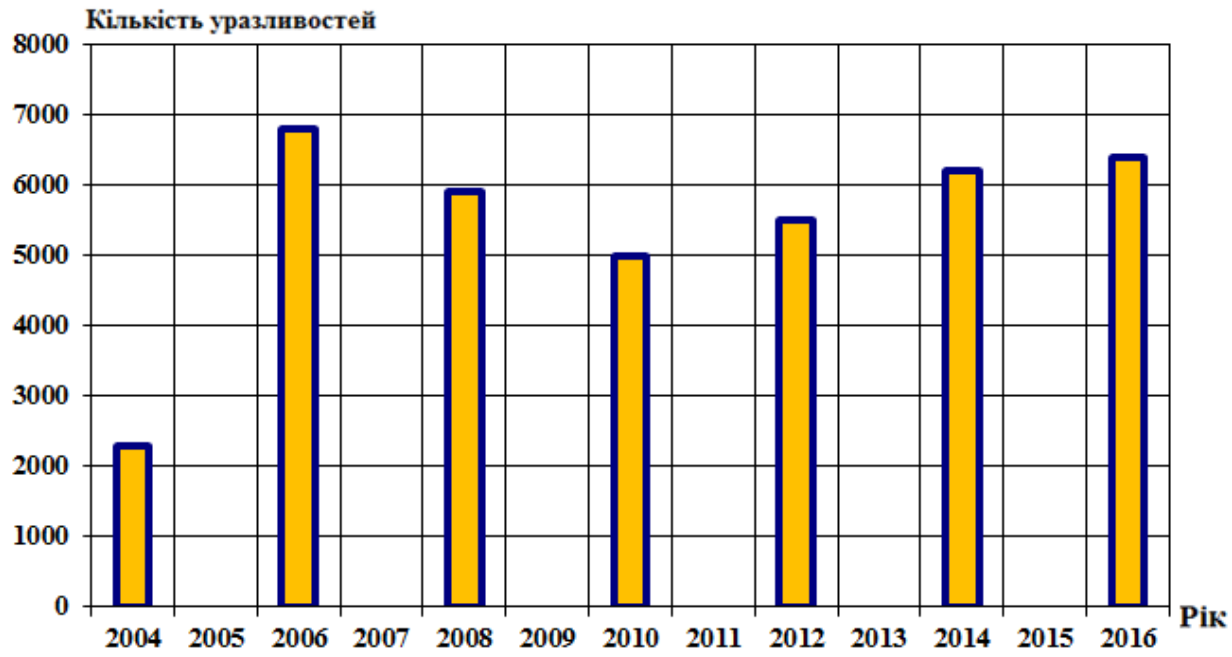


Рис. 1.3. Динаміка зростання уразливостей в КВІС, КВКС зв'язку та транспорту (Джерела: [93, 131, 138, 143, 174, 181, 186, 187, 206])

Стрімко зростає й кількість взаємопов'язаних вірусів, експлойтів та таргетованих атак, див. табл. 1.1, рис. 1.4.

Таблиця 1.1

Складні експлойти виявлені під час цільових атак на КВІС (Джерела: [93, 99, 103, 107, 143, 166, 181, 186, 187, 206])

Назва	**Рік**	**Платформа**	**Пов'язаний з ним**	**Тип**
1	**2**	**3**	**4**	**5**
Blue Termite	2013-2014	Windows	Немає жодного	Backdoor
Sofacy	2008-2014	Windows, Linux, iOS	MiniDuke	Trojan, Backdoor
MiniDuke	2008-Лютий 2013	Windows	CosmicDuke, Sofacy CozyDuke	Backdoor
CosmicDuke	квітень 2012- 2013	Windows	CozyDuke MiniDuke	Backdoor

продовження таблиці 1.1

1	2	3	4	5
CozyDuke	Липень 2014-2015	Windows	CosmicDuke, MiniDuke	Backdoor, Dropper
Equation	2002-2014	Windows	Flame - Stuxnet	Complex cyberattack platform
Flame	2007-Май 2012	Windows	Duqu-Gauss-Duqu 2.0 Equation-miniFlame Stuxnet	Complex cyberattack platform
Adwind	2012-2013	Windows, Linux, OS X, Android	Немає жодного	Backdoor, Complex cyberattack platform
Winnti	2009-2012	Windows	Немає жодного	Trojan
Cloud Atlas	2014-Серпень 2014	Windows, Android, iOS, Linux	Red October	Trojan
Red October	2007- Січень 2013	Windows, Windows Mobile	Cloud Atlas	Complex cyberattack platform
The Mask / Careto	2007-2013	Windows, OS X	Немає жодного	Cyberespionage toolkit
Naikon	2009-2011	Windows	Немає жодного	Trojan, Backdoor, Remote administration

Порушення працездатності КВІС, ІС або АСК у результаті кібератак вже багаторазово фіксувалось аналітиками, які займаються збором статистики по різним кіберінцидентам, табл. 1.2. Проте проблематиці кіберзахисту подібних систем з боку розробників все ще не приділяється достатньої уваги [168, 209]. Зокрема, ще наочно демонструється на різних щорічних конкурсах хакерів, Choo Choo Pwn (Південна Корея, 2013–2016 р.), Лас-Вегас (США, 2014–2016 р.) [80, 187].

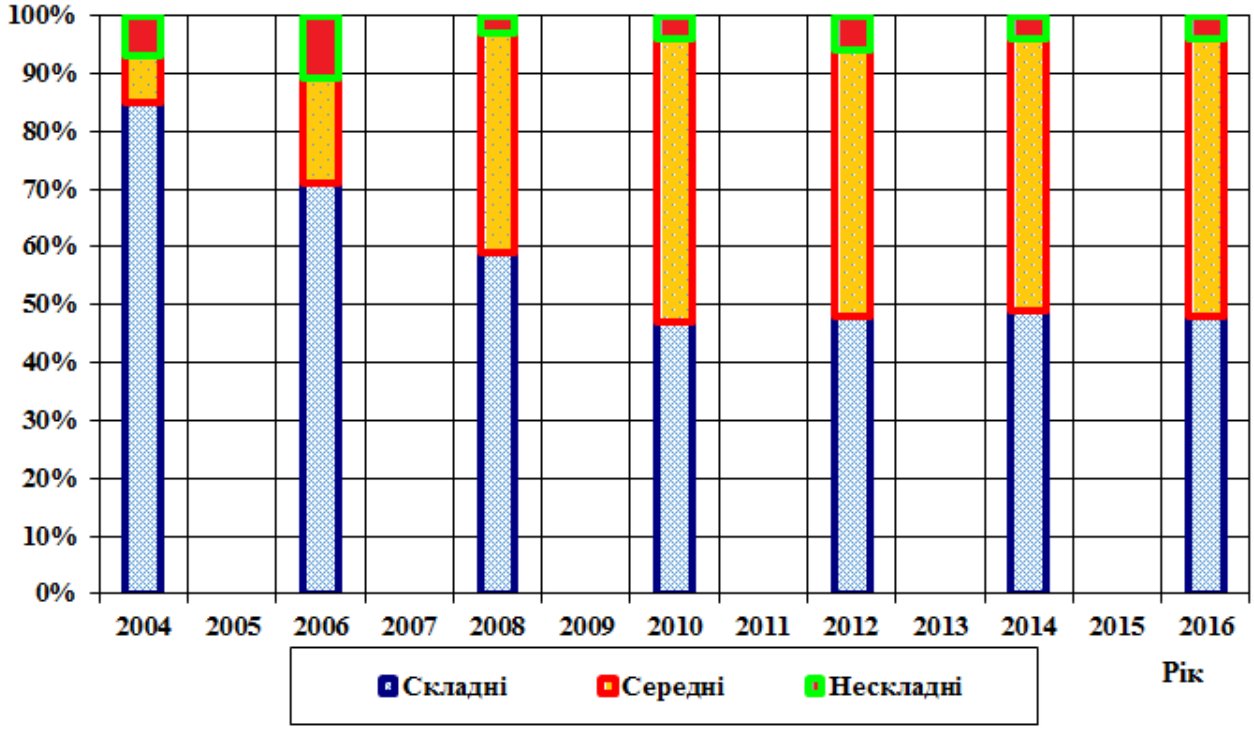


Рис. 1.4. Необхідна складність кібератак на КВІС та КВКС
(Джерела: [93, 131, 138, 143, 174, 181, 186, 198, 187, 206])

Існуючи уразливості КВІС, КВКС, SCADA, HMI, PLC обумовлені відсутністю механізмів захисту промислових протоколів, уразливостями ПЗ, помилками конфігурації обладнання та ін., а також, логікою бізнес-процесів, потребуючих підключення до зовнішніх мереж (корпоративних, WAN, Інтернет). Активний розвиток бездротових мереж й відкритих інформаційних технологій – ОС, мережевих протоколів і служб та ін., також не сприяють кібербезпеці критично важливих інформаційних систем та критично важливих комп'ютерних систем, які зараз активно впроваджуються у процеси управління багатьох державних установ та підприємств, а також приватних компаній.

Таблиця 1.2

Факти втручання в роботу КВІС різних галузей

№	Год	Держава	Подія	Наслідки
Зв'язок та Інтернет				
1	2002	Великобританія	НСД до службового телефонного зв'язку залізниці і системи управління семафором.	Відключений від зв'язку диспетчерський пункт залізниці, збій в системі включення семафорів [174, 181].
2	2014	США-Сирія	Хакери відключили 84 блоки сирійських IP-адрес	У результаті атаки вся Сирія втратила зв'язок з інтернетом на 3 доби [174].
Місцеві комунікації				
3	2011	США	Хакери відключили комп'ютери системи водопостачання в м. Х'юстоні, штат Техас.	Втрата контролю над роботою насосів, які керують водопостачанням на кілька годин [174, 187].
4	2011	США	Хакери відключили комп'ютери системи водопостачання в м. Спрінгфілд, штат Іллінойс.	Втрата контролю над роботою насосів, які керують водопостачанням на кілька годин [187].
Транспорт				
5	2003	Швеція, Ґетеборг	Проникнення в АСК рухом міських автобусів і таксі.	Втрата контролю над графіком руху на кілька годин [53].
6	2014	РФ	Вірус відключив веб камери фіксації швидкісного режиму "Стрілка-СТ" в Москві і області.	Камери виведені з ладу на кілька днів [168].
7	2003	США	Вірус SQL Slammer порушує роботу АСК авіакомпанії «Continental Airlines».	Скасування рейсів [171, 211].
Енергетичний сектор				
8	2009	США	Виявлено шкідливий код в комп'ютерах енергогенеруючих компаній	На протязі декількох місяців збиралася конфіденційна інформація [179].
9	2010	Іран	Виявлено вірус Stuxnet в системах енергосектору	Кілька місяців здійснювався кібершпіонаж за ядерною програмою Ірану та здійснено втручання в роботу критично важливого обладнання [179, 181].

Продовження таблиці 1.2

10	2011	США	SQL ін'єкція акаунтів VPN компаній енергетичного сектору	Кибершпіонаж [181].
11	2012	Іран	Атаки на АСК нафтового комплексу	Робота АСК припинена на дві доби [138, 143].
12	2012	РФ, Нидерланди, Велико-британія, США	Хакерська група Anonymous здійснила кібератаки на сервері великих світових нафтогазових компаній - «Газпром», «Роснефть», Shell, BP Global і ExxonMobil	У вільному доступі опинилися тисячі поштових акаунтів співробітників даних компаній [138, 143].
13	2013-2014	Сомалі, США, Велико-британія, Норвегія та ін.	Оприлюднено звіт компанії Rapid7 про факти втручання з боку злочинних угруповань в роботу GPS систем нафтовидобувних платформ, танкерів і контейнеровозів в Перській затоці і Аденській протоці.	Зафіксовані факти виходу з ладу ПЗ на бурових платформах на 19 діб [138, 143].
14	2015	Україна	Хакерська атака на електро розподільні підстанції	Відключення від електрозабезпечення декількох районів в захід-них областях України.

До теперішнього часу в світі склалася чітка система концептуальних поглядів на забезпечення ІБ та кібербезпеки держави в цілому та господарюючих суб'єктів, зокрема. Проте специфічні особливості КВІС та КВКС в різних установах не дозволяють створювати єдині універсальні методи забезпечення їх ІБ, в тому числі на етапі розпізнавання кіберзагроз або кібератак [1, 17, 93, 131, 138, 143, 174, 181, 186, 198, 187, 206].

Рішення задач забезпечення ІБ та кіберзахисту може бути отримано на базі використання різних інтелектуальних систем та технологій (ІСТ) в області розпізнавання кібератак, спрямованих проти КВКС та КВІС. Для цього багатьма фахівцями в області захисту інформації, запропоновано використовувати потенціал наступних систем: експертних (ЕС) [177]; підтримки прийняття рішень (СППР) [74, 111, 127], розпізнавання вторгнень (СРВ) [67, 84] або кібератак (СРКА) [74, 142, 169]. Використання подібних ІСТ дає позитивний ефект з ряду причин. По-перше, з'являються можливості

розв’язувати слабко формалізовані задачі із залученням нових, спеціально розроблених для цих цілей математичних моделей та методів, які ґрунтуються на використовуванні фреймів, семантичних мереж, теорії булевої та нечіткої логіки та ін. По-друге, ІСТ в області кіберзахисту та ІБ, орієнтовані на використання широким колом фахівців (останні використовують під час експлуатації зрозумілу багатьом фахівцям термінологію й техніку міркувань). По-третє, використання подібних систем, дозволяє значно підвищити результативність роботи СРКА, а також, оперативність прийняття рішень за рахунок акумуляції досвіду та знань багатьох висококваліфікованих експертів.

На думку багатьох фахівців [16, 23, 25, 33, 42, 63, 74, 88, 102, 106 та ін.], одним із найбільш перспективних шляхів підвищення результативності розпізнавання ОР у СРКА, зокрема й у випадках коли системи функціонують за умов апріорної невизначеності та впливу зовнішніх дестабілізуючих неконтрольованих факторів, є впровадження інтелектуальних інформаційних технологій (ІІТ), основаних на методах та моделях машинного навчання [9, 18, 19, 21, 49, 70, 77, 87, 113, 149, 164, 184]. На початковому етапі навчання, можуть бути використані методи та алгоритми кластерного аналізу [26, 50-52, 94, 95, 132, 148, 151, 164, 188, 190] для автоматизації процедури формування керованого процесу вхідного математичного опису задачі розпізнавання кібератак на основі бази знань (БЗ), сформованої по відомим ознакам.

Зазначимо, що трансформувати апріорно нечітке розбиття реалізацій ознак O_b кібератак (ОР) в чітке розбиття, на еквівалентні класи, для подальшого розпізнавання відомих та нових кібератак можна досить ефективно здійснити на основі використовування моделей, в якій контейнери (КОН) класів кібератак відновлюються в радіальному базисі бінарного простору реалізацій ознак конкретного об’єкту розпізнавання [26, 50-52, 94, 132, 148, 151, 164, 188, 190]. Наприклад, аналіз аномалій виявляє істотні відхилення трафіку мережевих пристроїв від «нормального» профілю

трафіку для даного пристрою або групи пристроїв. Прикладом мережевої аномалії є раптове збільшення інтернет-трафіку, зокрема, окремої робочої станції або зміна конфігурації пакетів або їхньої швидкості надходження (зокрема, збільшення SSL-трафіку) в порівняні із середньо статистичним показником для даної ЕОМ (АРМ) у складі КВІС [20, 23, 57, 75].

Інформація, яка приймається як базис для побудови класифікаторів кіберзагроз або кластерів для адаптивних систем розпізнавання кібератак (АСР), може бути подана в різних формах, наприклад, у вигляді реалізацій ознак аномалій в поведінці системи (ІС, АСК, КВІС, КВКС), кібератак або загроз для ІБ. В якості таких показників або метрик [75, 88, 104, 118] можна використовувати: порогові значення параметрів вхідного і вихідного трафіку; непередбачені адреси пакетів; атрибути запитів до баз даних (БД) і т. д. Для складних цільових атак інформаційні ознаки можуть бути досить нечіткими [28, 48, 53, 64, 70, 71, 100, 108, 117, 129, 134]. Наприклад, в ході складної кібератаки в кінці грудня 2015 року на КВКС енергосистеми України в Івано-Франківській області черговий комп'ютерного центру електропідстанції побачив, як стрілка курсору на дисплеї змістилася, хоча сам він не торкався до мишки. Курсор перемістився до віртуального перемикача, який відповідає за фізичний перемикач і перемкнув його. При цьому черговий оператор не зміг зайти в систему. Як показало розслідування, атака готувалася протягом тривалого періоду (щонайменше півроку). Зловмисники спочатку завантажили на комп'ютери підстанції програму Blackenergy 3, а потім шкідливу програму, яка взяла на себе управління електропідстанціями. Крім впровадження вірусу, атакуюча сторона запустила лавиноподібний потік дзвінків в Call-центр «ПАТ Прикарпаттяобленерго», щоб жителі не могли повідомляти про перебої з подачею електроенергії. Одночасно було відключено 30 електропідстанцій.

Квінтесенція процедури розбиття об'єктів, які охоплюють реалізації ознак кібератак на однотипові (з точки зору ІБ або кіберзахисту) класи множин, полягає в розбитті множини об'єктів на підмножини. Класичний

підхід до вирішення цього завдання здійснюється наступним чином [140, 154]. Спочатку визначається множина показників та їх параметрів – простір реалізацій ознак кібератак, див. табл. 1.3.

Таблиця 1.3

Фрагмент навчальної матриці простору ознак кібератак на КВІС

Тип кібератаки Джерела:[5, 31, 35, 58-61, 85, 98, 114, 115] Відомі загрози (Класи атак)	**Ознаки кібератаки** (Простір реалізацій ознак O_b) Джерела: [107, 146, 152, 162, 175, 184]
Відмова в обслуговуванні елементів КВІС та КВКС.	1 - не працюють штатні компоненти; 2 - зниження продуктивності системи; 3 - ін.
Викрадення інформації або компонентів КВІС та КВКС.	1 - об'єктивні ознаки (наприклад, поява конфіденційної інформації у ЗМІ); 2 - суб'єктивні ознаки; 3 - ін.
Привласнення особистості у КВІС та КВКС.	1 - об'єктивні ознаки (наприклад, зафіксовані спроби роботи під чужим логіном); 2 - суб'єктивні реалізації ознаки; 3 - ін.
Модифікація інформації у КВІС та КВКС.	1 - зміна контенту; 2 - зміна структури інформаційних масивів; 3 - ін.
Вірусна атака на КВІС та КВКС.	1 - незвичайні прояви в роботі ЕОМ; 2 - зміни заданої в передостанньому сеансі роботи з ЕОМ структури файлової системи; 3 - ін.
Несанкціонований запуск ПЗ КВІС та КВКС.	1 - незвичайні прояви в роботі ЕОМ; 2 - атипова поведінка ПЗ; 3 - ін.
Порушення доступності БД та ПЗ КВІС та КВКС.	1 - не працюють штатні компоненти КВІС (ІМ та ПЗ); 2 - ін.
Мережеві атаки.	1 - незвичайний трафік; 2 - аномалії мережевого трафіку; 3 - ін.
Інші.	
Невідомі загрози (Класи атак)	**Невиявлені реалізації ознаки кібератаки**

Таким чином, можна зробити припущення, що для підвищення ефективності процедури розпізнавання, множину значень кожного з показників доцільно за певними правилами розбивати на непересічні групи. У цьому випадку за кожним показником можна закріплювати специфічні виділені області його значень. Крім того, в межах цих значень, вимоги до стану ІБ (кіберзахисту) лишаються незмінними. На жаль, процедура розбиття реалізацій ознак для різних КВІС не є однаковою, що диктується специфікою їхньої роботи та функціональними завданнями. Наприклад, процедура розбиття реалізацій ознак кібератак на підгрупи для енергетичної компанії або транспортного підприємства не є однаковою. Розбиття проводиться не по одному, а по деякій множині показників захищеності конкретної КВІС. Отже, розбиття множини реалізацій ознак на непересічні групи проводиться не по одному, а по багатьом показникам. Далі потрібно визначити функцію близькості й критерій розбиття на множини реалізацій ознак з використанням великої кількості показників та їх значень, а також, окреслити коло та кількість класів типових кібератак на КВІС.

Дане завдання може бути розв’язано шляхом використання методів та моделей кластерного аналізу [26, 50-52, 94, 132, 148, 151, 164, 188, 190]. В традиційних алгоритмах нечіткої кластеризації об’єктів розпізнавання (ОР), як правило, використовуються не лише вхідні параметри які задають кількість кластерів розбиття, а й задані показники нечіткості кластерів в просторі реалізацій ознак ОР (наприклад, для уразливостей [10, 32], аномалій [7, 137, 150, 161, 179], загроз НСД [17, 36, 60] та кібератак [23, 153, 160]).

Отже використовуючи, інформаційний критерій функціональної ефективності (ІКФЕ) [94, 132], можна з часом реалізовувати в СРКА для КВІС адаптивні механізми налаштовування алгоритмів кластеризації реалізацій ознак ОР. Реалізація даної концепції дозволить оптимізувати в інформаційному розумінні кількість реалізацій ознак для ОР й отримати систему вирішальних правил [160, 191, 210], які дозволяють оперативного реагувати на кіберзагрози в робочих режимах експлуатації СРКА.

1.2. Огляд та аналіз методів які використовуються для побудови систем інтелектуального розпізнавання кібератак

Недосконалість існуючих методів захисту інформації, а також поява нових технологій та сценаріїв кібератак, можуть привести сучасні комп'ютерні системи у небезпечний стан. Тому необхідно своєчасно виявляти кібератаки і запобігати їхньому подальшому розвитку.

Існує багато робіт, які представляють собою огляди проблематики створення СРКА. Зокрема, роботи [7, 52, 137, 150, 161] присвячені аналізу методів виявлення аномалій у інформаційних системах та мережах. Автори пропонують схожі принципи класифікації цих методів, які базуються на машинному навчанні, статистичному аналізі, а також, методах на основі баз знань (БЗ).

Роботи [179, 200, 205] присвячені огляду методів машинного навчання для виявлення аномалій у ІКС.

Одними з перших у СРКА були використанні алгоритми побудовані на кінцевих автоматах (КА). Метод на основі КА моделює атаки у вигляді взаємопов'язаної мережі із станів і переходів [160]. Вторгнення вважається успішно реалізованим, якщо послідовність дій атакуючого переводить систему із стійкого стану в скомпрометований. Однією з перших спроб в реалізації цього підходу вважається робота [156]. Іншою системою виявлення зловживань на основі станів є IDIOT (Intrusion Detection In Our Time). Сценарії атак кодуються в шаблони IDIOT, а далі, події пов'язані із ІБ, перевіряються шляхом їх зіставлення з цими шаблонами [167]. Перевага даного методу полягає в можливості візуалізації переходів за допомогою діаграм переходу станів, а також у здатності системи виявляти атаку до її фактичної реалізації. З недоліків можна відзначити складність алгоритму виявлення нових атак для яких відсутні шаблони [167, 193, 200, 202].

Роботи [109, 200] присвячені принципам створення експертних систем (ЕС) для завдань розпізнавання кібератак та аномалій. Функціонування ЕС

базується на використанні продукційних правил. Продукційне правило має наступний вигляд [109, 200]: *IF condition THEN action*. У посилці даного правила міститься логічна умова, необхідна для здійснення атаки. Якщо всі умови в лівій частині правила виконуються, то, відповідно, далі реалізуються дії, зазначені в правій частині. Передбачається, що перед використанням ЕС адміністратор з інформаційної безпеки задає необхідні правила для СРКА. Подібні системи дають можливість використовувати людський досвід в СРКА або СІРКЗ [200].

Роботи [18, 27, 67, 71, 109, 110, 114, 176, 192] присвячені проблематиці використовування у СРКА методів обчислювального інтелекту та штучних нейронних мереж (ШНМ). Для навчання ШНМ для СРКА існує декілька методів. В роботі [166] представлено 12 алгоритмів для навчання ШНМ. У роботах [110, 192] для виявлення вторгнень використовується багатошарова нейронна мережа з двома прихованими шарами та вихідним шаром, якій містить три нейрони. В якості навчальної та тестової множини була обрана база даних DARPA 1998 (далі DARPA). Побудований на даній вибірці класифікатор був навчений розпізнавати два типи атак та нормальне з'єднання (*Normal*). В обох роботах для навчання нейромережі використовується алгоритм зворотного поширення помилки.

Іншою роботою, в якій використовується ця ж база даних, є [210]. Автор пропонує використовувати архітектуру багаторівневої нейронної мережі, в якій кожен з трьох рівнів містить окремий багатошаровий перцептрон. Розподільний шар кожного перцептрону складається з 30 нейронів. На кожному рівні здійснюється уточнення класифікації з'єднання. Так, на першому рівні визначається, чи є дане з'єднання атакою. Другий та третій рівні потрібні для визначення, відповідно, класу і типу атаки. Особливістю даного підходу є можливість отримання необхідного ступеня деталізації при класифікації розглянутого з'єднання.

Роботи [18, 27, 192, 210] присвячені виявленню аномалій із використанням нейронних мереж на основі даних, узятих з системного

журналу аудиту та лог-файлів окремих додатків. У [18] для завдання профілю кожного користувача застосовуються набори з найбільш поширених команд й частота їх використання. У [27] опрацьовувалася системна інформація, зокрема, використання системних ресурсів, час роботи та ін.

Винахід радіально-базисних нейронних мереж (РБНМ) та рекурентних нейронних мереж (РНМ) дозволив включити елементи пам'яті в моделі ШНМ. У [192, 210] наводиться огляд робіт із застосуванням РБНМ до завдань виявлення вторгнень. У [192] автори використали дану модель для передбачення наступних послідовностей системних викликів.

У роботах [177, 183] пропонується використовувати карти Кохонена для виявлення аномалій. З цією метою було зібрано достатньо даних, що характеризують легітимну поведінку користувачів.

У [185] карти Кохонена використовуються для попереднього опрацювання та кластеризації даних про трафік мережі. Опрацьовані дані дозволяють аналітику виявити зловживання у мережі.

В [210] для класифікації записів з набору даних DARPA використовується РБНМ, в якій перші два шари будуються за допомогою карт Кохонена. Результати експериментів показали, що дана модель має помітно кращі показники класифікації кібератак в порівнянні з базовим набором даних DARPA.

Роботи [170, 176, 182, 207] присвячені розгляду можливостей створення СРКА на принципах роботи генетичних алгоритмів. Як правило, генетичні алгоритми у СРКА використовуються у сукупності з іншими моделями класифікації даних: деревами рішень, елементами нечіткої логіки, нейронними мережами та ін.

У роботі [182] запропоновано використовувати генетичні алгоритми для створення нових правил розпізнавання зразків аномальних з'єднань у ІС. Навчальна вибірка містить 8 класів атак, тестова множина складається з 10000 записів, що описують 10 класів атак. Кожне правило представляється у вигляді дерева.

У [170] генетичні алгоритми використовуються для створення початкових популяцій нейронних мереж. Авторами запропоновано кодувати інформацію про вагові коефіцієнти нейромережі як двійкову послідовність. Коли генетичний алгоритм завершує роботу, створюється група нейронних мереж з найбільш виграшною структурою нейронів, кількістю прихованих шарів та початковим налаштуванням вагових коефіцієнтів.

У [207] застосовується нечітка логіка спільно з генетичним алгоритмом.

У [176] автор використовує правила класифікації виду *if condition then act* для розпізнавання шаблонів легітимного мережевого трафіку та зразків аномальних з'єднань. Посилка правила включала дев'ять атрибутів набору даних DARPA, які містять закодовані дані у вигляді рядка з 57 цілочисельних значень.

В [147] для генерації правил ЕС та СРКА задіяні генетичні алгоритми й дерева рішень. Для класифікації з'єднань використовувалися 5 атрибутів: IP адреси відправника і одержувача, їх порти та тип протоколу.

В [182, 210] генетичні алгоритми використовуються для генерації правил системи Snort на основі даних KDD Cup 1999. Результати експериментів доводять збільшення ефективності виявлення шаблонів атак і зниження обсягу задіяних для розпізнавання обчислювальних ресурсів.

Робота [187] присвячена розробці класифікаторів на основі формальних граматик, описаних за допомогою нотацій Бекуса-Наура. Автор наводить приклад, в якому кожне правило класифікації виглядає як ланцюжок умовних операторів або як закодована послідовності 8-бітових значень.

У роботах [157, 166, 185, 210] аналізується можливість створення СРКА на основі алгоритмів імунних систем. Цей підхід передбачає, що інформація про легітимність трафіку безпосередньо використовується для створення імунних детекторів. Кількість «антитіл» дорівнює кількості елементів, що описують легітимний мережевий трафік. Цей метод простий та

ефективний у виявленні аномалій. Проте, автори зазначають, що збільшення числа детекторів може привести до збільшення обчислювальних витрат у випадках розпізнавання складних аномалій та кібератак.

Запропонована авторами [166] система LISYS (Lightweight Immune System) призначена для виявлення вторгнень в розподіленому середовищі. Хоча ця система має ряд переваг, включаючи відносно невеликі обчислювальні витрати, надійність і масштабованість, автори відзначають деякі з недоліків. Серед них – неможливість виявлення мережевих атак із застосуванням протоколу UDP і можливість обійти систему шляхом проведення розподілених в часі атак типу – повільне сканування.

Роботи [144, 179, 200, 205] присвячені методам машинного навчання у СРКА. У роботі [144] автори запропонували замінити стандартний модуль виявлення в системи Snort деревами рішень. Експерименти були проведені на наборі даних DARPA та показали збільшення швидкості обробки PCAP-файлів в середньому на 40,3% в порівнянні із стандартним модулем.

У роботі [200] наведено порівняльний аналіз ефективності розпізнавання зразків даних DARPA 1998 за допомогою методу опорних векторів і ШНМ з одним прихованим шаром.

У роботах [155, 211] розглянуті можливості використовування у СРКА байєсовських мереж. У роботі [155] автори використовують оціночні функції для визначення апріорних та апостеріорних ймовірностей нових атак. Згідно [211] байєсівська мережа – це модель, яка кодує імовірнісні відносини між змінними, що описують стани системи.

Протягом багатьох років у СРКА використовувалися методи статистичного аналізу [74, 76, 103, 104, 133, 145]. Дані методи дозволяють опрацьовувати інформацію, отриману за деякий період часу, протягом якого поведінка системи або користувача є нормальною. Будь-яке значне відхилення значень параметрів, що характеризують системну або користувацьку діяльність, від порогових значень, заданих адміністратором, вказує на наявність аномалії.

У статистичних системах важливу роль відіграє правильний вибір контрольованих параметрів. Може вийти так, що через помилковий вибір кількості цих параметрів модель опису поведінки суб'єктів в системі виявиться неповною або надлишковою. Це призведе до ситуації коли атака не розпізнається або система спрацює помилково. Перевагами статистичних СРКА є можливість їх адаптації до зміни поведінки користувача, а також здатність виявляти невідомі типи кібератак.

Зазначимо, що зміст усіх статистичних методів навчання систем виявлення вторгнень (СВВ), полягає у знаходженні роздільної функції (РФ) [136, 144, 190, 203]. У свою чергу, алгоритм побудови РФ задає спосіб розбиття простору реалізацій ознак які використовуються під час розпізнавання [137, 142, 147, 205]. Основним недоліком статистичних методів для СВВ є неспроможність побудувати чіткі вирішальні правила за умови апріорного перетину класів об’єктів, що підлягають розпізнаванню. А перевагою є відносна простота та здатність здійснення статистичної корекції в процесі навчання СРКА.

В багатьох дослідженнях розглянуті можливості використання сучасних технологій обміну даними між різними БД СВВ. Наприклад, у працях [31, 39, 53] авторами розглянуті переваги технології клієнт-сервер для синтезу БЗ для СВВ. У роботах [25, 84, 70, 199] проаналізовано деякі аспекти використання методів та моделей для СРКА які базуються на застосуванні апарату нечіткої логіки з метою надання системам властивостей оперативності та адаптивності. Слід відмітити роботи у галузі створення СВВ з використанням різних підходів та методів штучного інтелекту. Наприклад, у працях [110, 166, 192, 210] розглядається використання нейромережевих технологій для вирішення завдань розпізнавання кібератак та аномалій. Але використання штучних нейронних мереж при складних випадках та таргетованих (цільових) кібератаках ускладнено через їх чутливість до багатовимірності БЗ реалізацій ознак. Також необхідна процедура попередньої нормалізації шаблонів атак [18, 27, 67, 192, 210].

У роботах [43, 80, 140, 173] авторами запропоновані певні підходи які дозволяють усунути недоліки детермінованого і статистичного методів розпізнання аномалій та кібератак. Запропоновано вирішальні правила для СРКА будувати за два кроки. На першому кроці будується найбільш просте вирішальне правило на підґрунті детермінованого методу. На другому кроці здійснюють корекцію цього правила. Зазвичай корекція реалізована за рахунок мінімізації кількості помилок першого та другого роду за класифікованою навчальною матрицею реалізацій ознак.

У роботах [176, 177, 183, 185, 199, 210, 211] авторами запропоновані гібридні підходи по побудови СРКА. В експериментальних системах використовуються три класифікатора: дерево рішень, метод опорних векторів (SVM – Support Vector Machines), а також, комбінація перших двох класифікаторів. Робота гібридного класифікатора складалася з двох фаз. Спочатку тестові дані подавалися на вхід вирішальних дерев. Далі тестові дані опрацьовуються за допомогою SVM, який видає результат класифікації. Ключовою ідеєю у використанні цього підходу було дослідження того, наскільки вузлова інформація від дерева рішень поліпшить ефективність SVM.

В роботі [177] використовуються штучні імунні детектори та карти Кохонена. Під час роботи система відстежує мережеві з'єднання та для кожного з них формує вектор реалізацій ознак. Тому будь-який вектор, відмінний від своїх клітин, вважається аномальним і подається на вхід СРКА.

В [185, 210] комбінується апарат імунних систем та нейронних мереж. В якості імунних детекторів використовуються багатошарові нейронні мережі. Експерименти були проведені на наборі даних KDD Cup 99, вони довели високу здатність детекторів адаптуватися до нових типів атак.

Іншим гібридним рішенням для СРКА є поєднання декількох нейронних мереж, які створюють єдиний класифікатор. Зокрема, робота [210] присвячена застосуванню методу SVM та РБНМ для класифікації записів з набору даних NSL – KDD. Підсумковий класифікатор являє композицію

послідовно побудованих на різних вибірках елементарних класифікаторів та процедури простого голосування.

Автори [176] пропонують об’єднувати вихідні значення від декількох нейронних мереж, які пройшли процес навчання за різними алгоритмами. На тестовій вибірці, що складається з 6890 зразків, досягнута точність класифікації близько 99%.

Стаття [199] описує дворівневу схему виявлення та класифікації кібератак. В дослідженні декілька адаптивних нейронечітких модулів були об'єднані разом. Кожен з них призначається для виявлення тільки одного класу атак та опрацьовує параметри записів KDD Cup 99. Підсумкова класифікація виконується нечітким модулем прийняття рішень, який реалізує систему нечіткого висновку з двома функціями належності.

Окрім розглянутих досліджень, існує багато робіт, що розкривають різні підходи до моделювання кіберзагроз для ІБ ІС та АС, зокрема, роботи які базуються на використанні: мереж Петрі [15, 22, 77, 159]; емуляції кібератак у послідовному й паралельному режимах [208]; причиново-наслідкової моделі [35, 89], концептуальних моделей кібератак [2, 34, 41, 73, 133, 144, 153, 166 та ін.], описових моделях мережі й зловмисників; моделюванні “виживання” комп'ютерних систем [182, 190], об'єктно-орієнтованому дискретному моделюванні [206]; моделі запит/відповідь [179, 200], ситуаційному моделюванні та ін.

У таблиці 1.4. представлені зведені дані про методи які використовуються у сучасних СРКА.

Однак як показав аналіз, сценарії кібератак постійно змінюються, наприклад, з появою нових технологій GSM, GPRS, GPS, LTE, Wi-Fi тощо, оскільки зловмисники використовують індивідуальні підходи, а також у зв'язку з регулярними змінами в ПЗ й апаратних засобах ІКС [72, 76, 112, 184, 190, 208].

Таблиця 1.4

Сучасні системи виявлення кібератак

Метод (Модель)	Рівень спостереження	Аномалії/Зловживання	Адаптивність	Обчислювальна складність	Використовується у СВВ	Контрольоване навчання	Розпізнавання комбінованих атак
Модель переходів	Hybrid	-/+	-	o(N)	DREM, JANUS	-	-/+
Графи атак	Hybrid	-/+	+	NP	Bro	-	-/+
Нейронні мережі	NIDS	+/+	+	o(N) та вище	Hyperview	+	-/+
Імунні мережі	NIDS	+/+	+	o(N) та вище	NSL-KDD IDS	-/+	-
Експертні системи	NIDS	-/+	+	NP	NIDES, DIDS	-/+	-/+
Статистичні методи	NIDS	+/-	+	o(N)	NSM, Haystask	-/+	-/+
Кластерний аналіз	Hybrid	+/+	+	o(N) та вище	FIRE, Y-means	-/+	-/+
Поведінкова біометрія	NIDS	+/-	+	o(N) та вище	MDS, KDS	-/+	-
Сигнатурні методи	Hybrid	+/-	+	o(N) та вище	Snort, Sentarus IPS	-/+	-
HIDS - спостереження на рівні ОС вузла мережі (або спостереження на рівні мережевої взаємодії об'єктів на вузлах мережі); **Hybrid** - комбінація спостерігачів різних рівнів							

В роботах [132, 148, 151, 199] запропоновано подальший розвиток методології застосування шаблонів (еталонів) для розпізнавання окремих класів аномалій та кібератак. З метою зменшення кількості базових шаблонів й спрощення процедури мінімізації покриття класу, основну увагу зроблено на редукцію простору реалізацій ознак розпізнавання об’єктів шляхом впровадження параметру оцінювання інформативності кожної реалізації ознаки.

Варто зазначити, що на думку багатьох дослідників, підходи до створення СРКА, які реалізують класичні алгоритми [60, 64, 69, 84, 110, 135, 170], підходять для заздалегідь визначених класів, загроз, аномалій та кібератак. У тому випадку якщо виникає нова уразливість та пов’язана з нею загроза, яка не блокується системами кіберзахисту (СКЗ), необхідно передбачити можливість зміни архітектури базової СРКА.

Незалежно від задіяних у КВІС (КВКС) СРКА, існуючи системи стикаються з однаковою проблемою – постійною зміною нападниками сценаріїв кібератак. Це, у свою чергу, вимагає розробки більш гнучких СКЗ, здатних залишатися ефективними, навіть якщо не відомі точні параметри кібератаки, а також її ознаки [7, 52, 60, 64, 69, 84, 110, 135, 137, 144, 170, 179, 200, 205]. Враховуючи надзвичайно велику різноманітність видів можливих атак на ІКС та критичність можливих наслідків, необхідні подальші дослідження з проблематики розробки адаптивної СРКА, здатної враховувати постійну зміну параметрів та реалізації ознак кібератак. На думку багатьох фахівців [176, 177, 183, 185, 199, 210, 211] подальший напрямок розвитку методів виявлення кібератаки бачиться у гібридизації існуючих підходів та створенні адаптивних СРКА, ЕС та СППР для завдань забезпечення належного рівня ІБ КВІС та КВКС.

Одним із класів адаптивних СРКА (АСРКА), теорія аналізу та синтезу яких все ще знаходиться на етапі теоретичних досліджень [73, 87, 151, 176, 177, 183, 185, 199, 210, 211], є знання-орієнтовані системи. В подібних АСРКА прийняття рішень здійснюється шляхом розпізнавання шаблонів загроз, аномалій та кібератак, синтезованих за якісними шкалами виміру відповідних реалізацій ознак.

Актуальним є також, напрямок розвитку теорії та практики застосування АСРКА для завдань планування та корекції політики інформаційної безпеки КВІС [8, 13, 16, 74, 139]. Вважаться, що подібні системи повинні мати доступ до попередньо накопичених знань. Це дозволяє здійснювати процедуру інтелектуального аналізу даних (Data mining) з метою підвищення ефективності й достовірності рішень СІБ [27, 188, 203].

На практиці АСРКА будуються за багатофункціональним принципом [177, 185, 210]. Це дозволяє здійснювати збір та накопичення інформації у БЗ, об’єднаних інформаційно-комунікаційним середовищем.

У загальному випадку СРКА та СППР які використовуються у завданнях кіберзахисту та інформаційної безпеки за типом вирішальних правил можна класифікувати наступним чином [94, 109, 183, 185, 199, 200]:

детерміновані СППР. Процедура прийняття рішень виконується за детермінованими вирішальними правилами без навчання. Подібні системи містять прості вирішальні правила, але відрізняються невисокою достовірністю рішень;

статистичні СППР. Для подібних систем процедура прийняття рішень здійснюється на підставі максимізації статистичних критеріїв функціональної ефективності систем;

детерміновано-статистичні інтелектуальні СППР. Подібні системи дозволяють поєднати всі позитивні властивості попередніх підходів до розв'язання задач аналізу загроз, аномалій та виявлення кібератак.

Інтелектуальні СРКА (або СІРКЗ) повинні відповідати сформованим в процесі еволюції систем виявлення вторгнень (СВВ) базовим принципам [53, 84, 94, 109, 135, 183, 185, 199, 200]:

достовірність та внутрішня несуперечливість вхідної інформації яка стосується станів об'єкту кіберзахисту;

адаптуємось до зростання кількості деструктивних впливів на ІБ;

зрозумілість інтерпретації на основі баз знань, які зокрема, містять опис реалізацій ознак кібератак;

можливість пояснення й обґрунтування прийнятих рішень з результатами розпізнавання певних загроз, аномалій в системі чи виявлених кібератак;

можливість переоцінки прийнятих рішень у разі повторної процедури розпізнавання.

Як показав аналіз багатьох досліджень основна відмінність СППР, які використовуються в завданнях забезпечення ІБ, від ЕС полягає в наявності зворотного зв'язку, який дозволяє ЕС самостійно сформувати БЗ. В ході цієї процедури навчання, здійснюється адаптація та ітераційна оптимізація

параметрів БЗ за рахунок відповідного критерію функціональної ефективності системи (КФЕС).

Суттєвою перевагою поєднання СРКА та ЕС (або СППР) є можливість їх застосування у випадках, коли особа, що приймає рішення має недостатній професійний досвід або при необхідності дати пояснення складних випадків нелегітимного втручання (НЛВ) у роботу КВІС та реалізаціями ознак, які потребують аналізу великих обсягів інформації. Адаптивна, здатна до самонавчання ЕС може використовувати інформацію накопичену у БЗ та прийняти рішення на основі більшої кількості фактів НЛВ ніж ті які аналізуються співробітниками служби інформаційної безпеки (СІБ).

Аналіз відомих рішень [1, 25, 70, 97, 176, 177, 183, 185, 199, 210, 211] показує, що основною тенденцію розвитку сучасних СРКА, ЕС та СППР для завдань ІБ, є надання їм властивості прогнозувати перебіг та наслідки цільових кібератаках, тобто можливість адаптуватися до нової інформації про стан систем, використовуючи методи машинного навчання та теорії розпізнавання. Однією з проблем, що зустрічаються при розпізнаванні складних цільових кібератаках є вимірність простору реалізацій ознак [75, 85, 102, 117]. Існує ціла група методів [50-52], які чутливі до розмірності вхідних даних і добре працюють з матрицями реалізацій ознак невеликої розмірності [141, 175, 179, 200], але у випадку коли інформаційні масиви мають велику розмірність, вони не завжди дозволяють отримати адекватні результати та вирішити поставлені перед СРКА завдання.

Основним завданням навчання АСРКА, ЕС або СППР є побудова безпомилкових вирішальних правил за відповідною навчальною матрицею реалізацій ознак – БВПНМ. На практиці побудова БВПНМ суттєво ускладнена у разі збільшення кількості реалізацій ознак які використовуються для наповнення алфавіту класів розпізнавання. У загальному випадку для вирішення цієї проблеми використовують елементарні ієрархічні класифікатори (ЕІК) [50-52, 133, 147, 166, 169]. Одним із підходів до побудови ЕІК є використання методів, описаних у праці [169].

Авторами досліджуються методи ієрархічної класифікації аномалій, загроз та кібератак за допомогою термінології яка притаманна графам атак. З методологічної точки зору, методи аналізу і синтезу адаптивних, інтелектуальних СРКА, ЕС або СППР в завданнях визначення ефективної ПБ, у межах детерміновано-статистичних підходів є прогресивними. Це обумовлено їхньою здатністю до реалізації когнітивного механізму прийняття рішень, якій зазвичай притаманний людині. Але у випадках якщо шаблони аномалій або кібератак належать до випадку коли є перетин класів об'єктів розпізнавання, подібні системи не завжди дають бажану точність [166, 169]. Одним із перспективних напрямів аналізу і синтезу АСРКА, ЕС, СППР, що навчаються за умов апріорної невизначеності, у рамках застосування детерміновано-статистичних моделей, є використання методів, моделей та алгоритмів машинного навчання (ІЕТН) [50-52, 162]. Основна ідея цього підходу ґрунтується на максимізації інформаційної спроможності ЕС, СППР або АСРКА за рахунок використовування під час навчання додаткових інформаційних обмежень [151, 200]. Зокрема, у рамках ІЕТН час машинного навчання АСРКА, ЕС або СППР, здійснюється трансформація апріорно нечіткого розбиття простору реалізацій ознак аномалій, кіберзагроз або кібератак на чітке розбиття. Це досягається за рахунок ітераційних процедур. Окрім цього, здійснюється цілеспрямований пошук глобального максимального значення багатоекстремальної функції в робочій області її визначення. Одночасно відновлюються оптимальні роздільні класи (КОН), що побудовані у радіальному базисі зазвичай бінарного простору реалізацій ознак об'єкта розпізнавання (ОР) [7, 20, 28, 35, 122, 127].

Таким чином, актуальними є подальші дослідження з вибору методів та моделей формування раціональної кількості реалізацій ознак для перспективних інтелектуальних АСРКА, ЕС та СППР для аналізу відповідного класу кіберзагроз, аномалій та кібератак. Вибір конкретного методу визначається його спроможністю перетворити вхідну інформацію

(наприклад, вхідні сигнали) у послідовність реалізацій ознак, які будуть унікальні для кожного класу об’єктів розпізнавання.

1.3. Висновки до першого розділу

Виконаний огляд і аналіз попередніх досліджень у сфері вирішення завдань захисту КВІС та КВКС, підвищення стійкості інформаційно-обчислювальних процесів, схоронності й захищеності інформації, дозволив зробити наступні висновки:

1) обґрунтовано, що використання існуючих підходів не повною мірою враховує специфіку побудови та експлуатації СКЗ, оскільки:

- не повністю в існуючих СРКА та СІРКЗ для КВІС та КВКС враховується поява нових класів кібератак, пов’язаних із поширенням систем та технологій супутникової навігації, відеоспостереження, Wi-Fi, GSM, GSM, VSAT, систем SCADA, HMI, PLC та ін., що не дозволяє проводити їх дослідження, здійснювати вибір раціональних способів протидії і нейтралізації наслідків, аналізувати складніші і раніше невідомі види цільових кібератак;

- не повністю досліджено загрози для інформаційної безпеки КВІС та КВКС при реалізації складних кібератак;

- недостатньо повно формалізовано задачі та методи побудови адаптивних інтелектуальних систем розпізнавання кібератак на КВІС та КВКС;

2) з’ясовано, що складність застосування до адаптивних інтелектуальних систем розпізнавання цільових кібератак формалізованого апарату аналізу й синтезу СІРКЗ, полягає в тому, що конкретний інформаційний комплекс КВІС або КВКС та їх підсистеми ІБ складаються з різнорідних елементів, які описуються із використанням різних моделей;

3) показано, що застосування елементів адаптивного захисту інформації може бути засноване на використанні новітніх методів інтелектуального розпізнавання кібератак на КВІС;

4) на підставі аналізу стану ІБ КВІС та КВКС в Україні формалізовано загальну наукову задачу досліджень, спрямованих на подальший розвиток моделей та методів захисту на основі інтелектуального розпізнавання кібератак в умовах збільшення кількості та складності дестабілізуючих впливів на конфіденційність, цілісність і доступність інформації;

5) формалізована мета роботи і групи задач, що її вирішують.

Таким чином, сформульована актуальна наукова задача подальшого розвитку моделей та методів захисту на основі інтелектуального розпізнавання кіберзагроз для державних та приватних установ в умовах зростання кількості дестабілізуючих впливів на інформаційну безпеку КВКС та КВІС.

Для розв’язання поставленої в роботі задачі необхідно вирішити такі завдання:

Для досягнення поставленої мети необхідно розв'язати такі задачі:

1) проаналізувати методи які використовуються для побудови систем інтелектуального розпізнавання кібератак в умовах зростання кількості та складності цільових атак, які характеризуються реалізаціями ознак, які важко пояснити, а також ураховують відомі статистичні параметри кластеризації реалізацій ознак кібератак;

2) розробити модель експертної системи та метод її навчання із використанням процедури нечіткої кластеризації реалізацій ознак кібератак та можливістю корекції вирішальних правил, що дозволить створювати адаптивні механізми самонавчання систем кіберзахисту;

3) удосконалити методи навчання експертної системи та визначити раціональну кількість кластерів у просторі реалізацій ознак кібератак, що дозволить зменшити час навчання;

4) розробити експертну систему розпізнавання кібератак (об’єктів

розпізнавання – ОР), яка у взаємодії із іншими системами кіберзахисту дозволить підвищити рівень розпізнавання складних цільових кібератак;

5) провести імітаційні дослідження та натурну апробацію моделі експертної системи.

РОЗДІЛ 2
МОДЕЛЬ АДАПТИВНОЇ СИСТЕМИ ІНТЕЛЕКТУАЛЬНОГО РОЗПІЗНАВАННЯ КІБЕРАТАК ІЗ ВИКОРИСТАННЯМ ПРОЦЕДУРИ НЕЧІТКОЇ КЛАСТЕРИЗАЦІЙ РЕАЛІЗАЦІЙ ОЗНАК

Сучасні кібератаки стали надзвичайно складними. Вузько направлені, систематичні і розподілені атаки, відомі як постійні складні загрози, здатні ховатися від антивірусів, не виявляються міжмережевими екранами та системами виявлення вторгнень. Термін «адаптація» в СІРКЗ має на увазі процес цілеспрямованої зміни структури, алгоритму або параметрів системи з метою підвищення ефективності її функціонування. Адаптивна СІРКЗ є системою зі зворотним зв'язком, класифікація адаптивності якої проводиться за трьома основними ознаками: рівнем адаптації (структурна, алгоритмічна, параметрична); механізмом адаптації (з еталонною моделлю; система з аналітичним налаштуванням; екстремальна система); способом адаптації (з дискретним або безперервним налаштуванням). Особливістю адаптивних систем виявлення та розпізнавання атак є наявність спеціальних алгоритмів, які забезпечують вирішення низки взаємопов'язаних завдань, таких, як збір і аналіз інформації про стан комп'ютерної системи; оцінка стану зовнішнього середовища; прийняття рішення про необхідність застосування заходів захисту; вибір керуючих параметрів; синтез схеми адаптації та реалізацію механізмів її реалізації; визначення ефективності системи захисту.

Основні вимоги до адаптивних алгоритмів в СІРКЗ для КВІС можна сформулювати наступним чином [27, 28, 42, 67, 80, 119, 210]: оперативність у часі, що пов'язано з необхідністю застосування результативних за швидкодією алгоритмів; найнижчий з допустимих рівень складності контуру адаптації; автоматичний або автоматизований характер процедури адаптації; мінімальне число керованих параметрів, що дозволить мінімізувати витрати часу і пам'яті, необхідні для реалізації алгоритму, при цьому забезпечити заданий рівень захисту.

Як було показано у попередньому розділі, одним із перспективних та актуальних напрямів досліджень систем інтелектуального розпізнавання кібератак є надання їм властивості адаптивності. Зокрема, в СІРКЗ можна застосувати моделі та методи інформаційно-екстремальної технології [26, 50–52, 80, 94, 67, 169, 200, 210], яка заснована на концепції підвищення функціональності СРКА за рахунок імплементації в процедуру навчання спеціальних інформаційних обмежень, які, наприклад, стосуються реалізацій ознак в межах відомих та нових класів кібератак.

2.1. Формалізована постановка задачі інформаційного синтезу адаптивної системи розпізнавання кібератак

Комп'ютерні системи та технологій, які входять до складу КВІС, зазвичай добре захищені. Для успіху атакуючий стороні потрібно вміти проходити або відключати захист. Зловмисники використовують унікальні, ще не відомі ІТ-індустрії шкідливі програми, уразливості і способи реалізації кібератак. Таким чином СІРКЗ (або АСРКА) повинні бути розраховані на безперервний процес навчання та поповнення бази знань (БЗ), рис. 2.1.

Більшість сучасних СРКА та СВА базується на моделях і методиках, заснованих на теорії розпізнавання образів [83, 99, 109, 140, 210]. Відповідно до базових принципів цієї теорії для виявлення кібератак, необхідно сформувати образ нормальної та аномальної поведінки КВІС (КВКС), наприклад, використовуючи думки та оцінки експертів. Сформований таким чином образ, може бути описаний як сукупність значень параметрів оцінки, тобто ознак. Для зручності можна застосувати двійкову форму опису реалізацій ознак, які зберігаються у репозиторії.

Механізми класифікації використовуються на початковому рівні, наприклад, для систематизації способів захисту (нечіткі висновки) по вектору нечітких реалізацій ознак кібератак. Якщо достовірність класифікації по відомим загрозам менше заздалегідь відомого рівня, то при наявності

реалізацій ознак атаки, класифікація розширюється за рахунок введення нової градації, тобто діє процес кластеризації об’єктів розпізнавання (ОР). Асоціації виявляють причинно-наслідкові зв'язки і визначають ймовірності або коефіцієнти достовірності (валідації), дозволяючи робити, наприклад, за допомогою ЕС відповідні висновки.

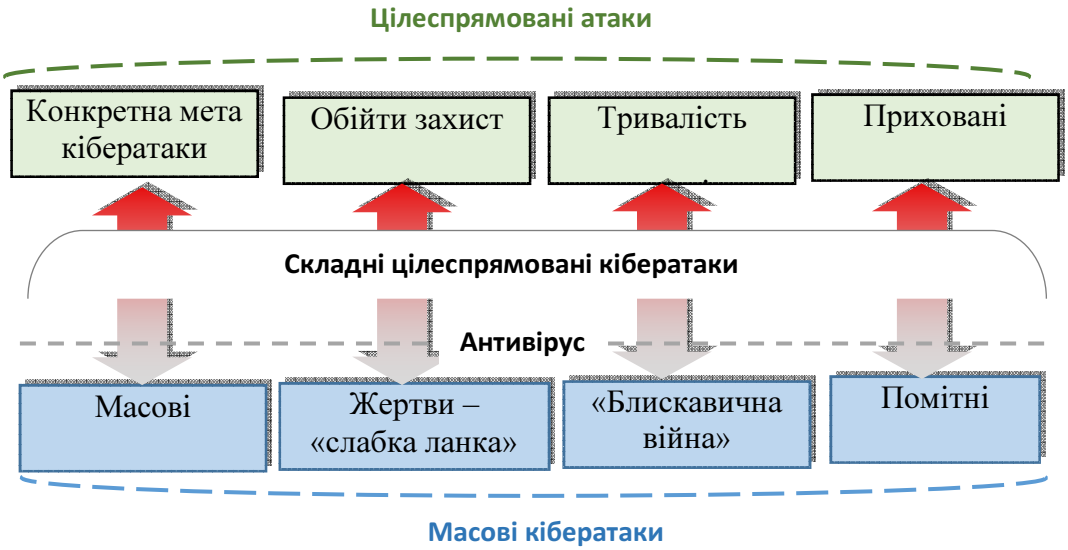


Рис. 2.1. Взаємозв’язок між типами атак які підлягають виявленню у адаптивних системах інтелектуального розпізнавання кібератак

Побудова моделі інтелектуального аналізу даних є частиною масштабного процесу опрацювання даних у АСР, в який входять всі завдання, від формулювання питань вибору та зберігання даних і створення моделі до розгортання моделі в робочому середовищі КВІС.

Складність імплементації в існуючі моделі СІРКЗ формалізованого апарату теорії розпізнавання, полягає в тому, що конкретний інформаційний комплекс для КВІС, що включає в себе часто унікальне ПЗ та відповідні інформаційні масиви, а також власну підсистему ІБ, складається з різнорідних компонентів. Ґрунтуючись на попередніх висновках розділу 1, в рамках дисертаційного дослідження розглядається СІРКЗ, що базується на

багатоетапному етапному виявленні загроз, аномалій і кібератак, див. рис. 2.2.

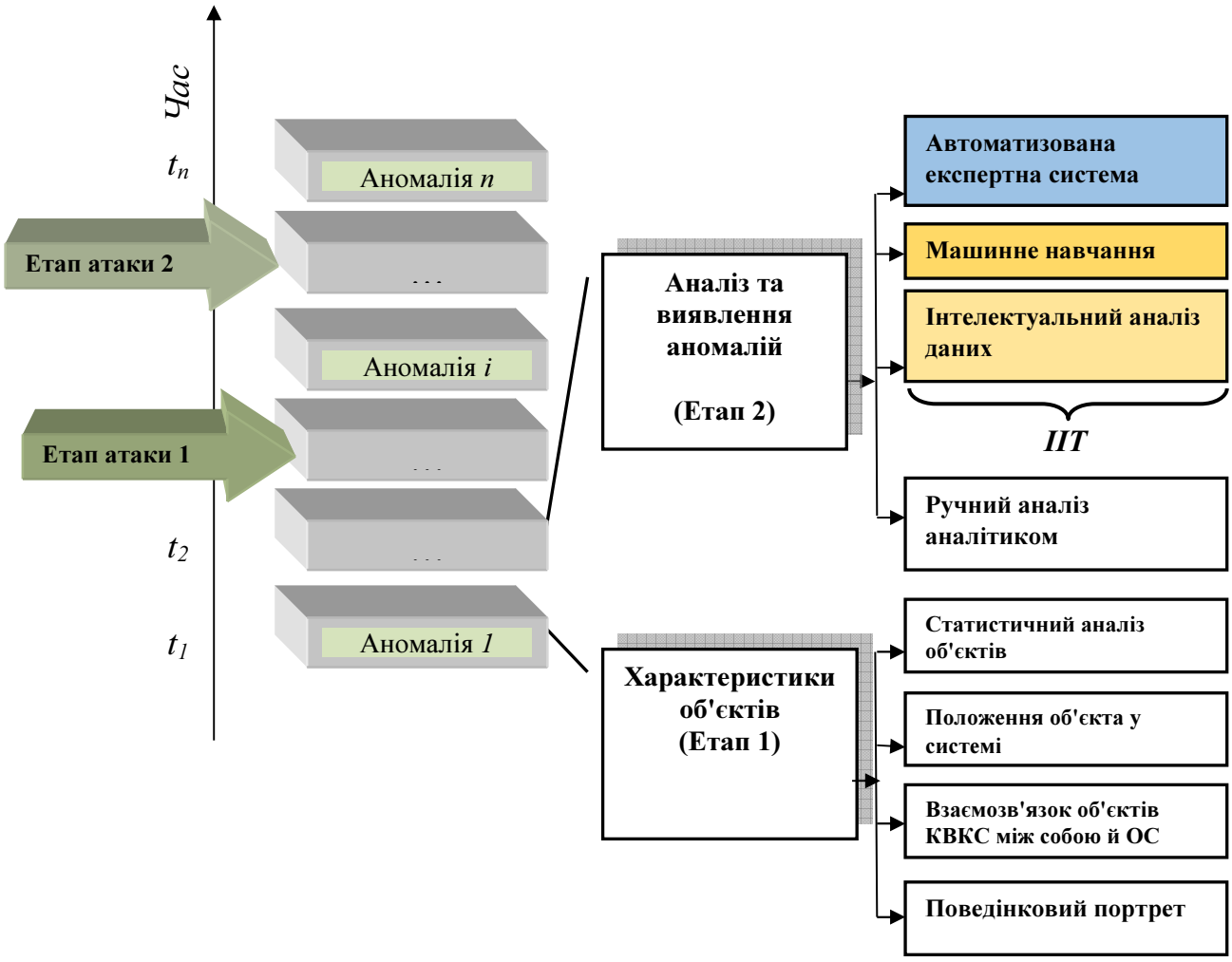


Рис. 2.2. Схема роботи багатоетапної СІРКЗ

Серед методів, використовуваних у СІРКЗ, можна виділити два напрямки: одні спрямовані на виявлення аномалій у системі, що захищається, а інші – на пошук зловживань [18, 19, 27, 28, 31, 49, 67, 75, 80, 104, 113, 156, 161, 203, 211]. Кожен зі згаданих напрямків має свої переваги та недоліки, аналізу яких присвячено багато робіт, зокрема [49, 67, 75, 80, 104, 113, 156]. В більшості сучасних СІРКЗ застосовуються комбіновані рішення, засновані на синтезі двох або трьох методів розпізнавання [80, 104, 113, 156]. При цьому, головна мета методів які використовуються для розпізнавання

аномалій – встановити, чи є процес, що викликав зміни в роботі КВІС (КВКС), діями атакуючих. У свою чергу, серед цих методів можна виділити дві групи: контрольоване (КН) та неконтрольоване навчанням (НН) СВВ. Основна відмінність між ними полягає в тому, що методи КН використовують встановлений набір реалізацій ознак об'єктів розпізнавання (ОР), а також, апріорно відомі значення цих ознак ОР, та, крім того, час навчання фіксований. У НН множину параметрів (ознак) можна змінювати протягом процесу навчання, наприклад, у випадках збільшення кількості загроз і уразливостей. Для НН процес навчання є постійним.

Метою другого напрямку (виявлення зловживань) є пошук подій (або послідовностей подій), що визначені (адміністратором з ІБ або експертом під час навчання СВВ) як етапи виконання кібератаки.

Для забезпечення високонадійного опрацювання даних у КВІС (КВКС) в умовах збільшення кількості деструктивних впливів, зокрема й кібератак, потрібно знайти:

$$SI^{*} = Arg \max_{CO^{ad} \in CO, CM^{ad} \in CM, ME^{ad} \in ME} SI\left[(CO^{ad}, CM^{ad}, ME^{ad})\right] | \Lambda, \quad (2.1)$$

де CO^{ad} – допустимі параметри регулювання КВІС (КВКС);

CM^{ad} – допустимі для можливого застосування методи і моделі протидії загрозам та кібератакам на основі СІРКЗ та ідентифікації стану КВІС;

ME^{ad} – допустимі для можливого застосування засоби попередження, виявлення, аналізу кібератак;

Λ – обмеження на параметри, від яких залежить критерій результативності АСР (потенційно вразливі місця КВІС, період часу дії кібератак, вартість засобів захисту та ін.).

Метод та модель інформаційного синтезу адаптивних та здатних до самонавчання СІРКЗ (ЕС або СППР для ІБ, далі по тексту – адаптивні системи розпізнавання – АСР), повинен відповідати наступним вимогам:

мати здатність корегувати математичний опис даних у рамках детерміновано-статистичного підходу, зокрема, на вході у СІРКЗ з метою побудови елементарних класифікаторів (ЕК) ОР;

містити набір ЕК для всіх ОР, який забезпечує під час тестових випробувань повну достовірність рішень, яка наближена до максимальної асимптотичної;

мати високу функціональну ефективність процедури навчання АСР, а також, об'єктивно характеризувати результативність систем кіберзахисту для КВІС;

мати здатність оптимізувати просторово-часові параметри під час опрацювання вхідних даних за інформаційною умовою функціональної результативності (ІУФР) навчання адаптивних ЕС (або АЕС);

володіти властивістю до складання прогнозу, що до зміни функціональної результативності та надійності АСР;

бути багатофункціональним для здатних до самонавчання АСР, а також, мати можливість бути імплементованим в рамках об'єктно-орієнтованих методологій проектування систем кіберзахисту.

Таким чином, інформаційний синтез адаптивної здатної до самонавчання СІРКЗ, полягає в оптимізації за ІУФР параметрів функціонування системи, які мають вплив на її результативність. У свою чергу, параметри АСР, визначаються як показниками якості процесу навчання, так й організаційно-методологічними обмеженнями в процесі формування БЗ та набору ЕК.

На відміну від традиційних математичних методів, які використовуються у СВВ, основною особливістю ІІТ в завданнях аналізу і синтезу АСР є застосуванні концепції інтелектуального аналізу даних в рамках евристичного підходу під час навчання [80, 113, 211]. Згідно з

теорією класифікаційного керування, структуру АСР можна подати як сукупність наступних взаємопов'язаних компонентів:

об'єкт керування – система інформаційної безпеки (СІБ) КВІС (КВКС).

аналітик(ки) з ІБ як керуючий орган системи кіберзахисту КВІС (КВКС);

ЕС (або СППР), яка ґрунтується на теорії машинного навчання та розпізнавання образів [49, 80, 190, 210]. Головна функція – є оцінка поточного стану ІБ за кіберзахисту КВІС (КВКС) та формування рекомендацій для служб які опікуються захистом інформації;

модуль попереднього опрацювання даних (МПОД). Головна функція МПОД – формування вхідних параметрів які дозволяють оцінювати стан об'єкту захисту під час реалізації складних кібератак;

модуль кінцевого опрацювання даних (МКОД). Головна функція МКОД – реалізація технологій нівелювання загроз або кібератаки, а також розробка схеми знешкодження наслідків для кожної кібератаки).

У рамках ІІТ, які використовуються для навчання систем розпізнавання нелегітимних дій атакуючої сторони, основним завданням АСР є ефективна процедура трансформації нечіткого розбиття простору реалізацій ознак кібератак у чітке розбиття класів ОР. Це, зокрема, досягається шляхом використовування ітераційної процедури яка дозволяє оптимізувати параметри функціонування АСР в завданнях підтримки високого рівня ІБ КВІС. Процес навчання проходить в два етапи:

перший етап передбачає цілеспрямований пошук глобального максимального значення багато екстремальної функції для статистичного інформаційного параметру в робочій області реалізацій ознак ОР;

другий етап дозволяє визначити й одночасно відновити оптимальні роздільні поверхні [5, 26, 50-52, 80, 166, 169], які були побудовані в бінарному просторі реалізацій ознак ОР.

Вхідний нечіткий розподіл реалізацій об'єктів, які використовуються в процесі навчання (ОВН), трансформується в чіткий розподіл під час

оптимізації перевірочних допустимих відхилень на кожен клас кібератак. У результаті відбувається цілеспрямована зміна значень реалізацій ознак розпізнавання у АСР для визначених об'єктів та побудова коректних вирішальних правил за багатовимірною бінарною навчальною матрицею (ББНМ).

Таким чином, у рамках ІІТ навчання АСР можливо поєднати процес корегування ОВН й безпосередній етап навчання. Під час останнього етапу відбувається синтез коректних вирішальних правил.

Як було показано у першому розділі роботи, одним із перспективних напрямів надання властивості адаптивності ІІТ та системам розпізнавання нелегітимних дій з боку атакуючих КВІС є використання методів та моделей та машинних технологій навчання СВВ (МІЕТ), які ґрунтуються на максимізації інформаційної спроможності СІРКЗ в межах відомих та нових класів вторгнень [50, 80, 94, 133, 149, 166, 169, 200, 210].

Розв'язання завдання по формуванню вхідного математичного подання адаптивної, здатної до самонавчання ЕС у складі АСР, полягає у створенні об'єкту який використовується для навчання – ОВН (тобто, навчальної матриці реалізацій ознак – learning matrix) – $\left\| lm_{m,i}^{(j)} \mid m=\overline{1,M}; i=\overline{1,N}, j=\overline{1,n} \right\|$. Потрібно розв'язати наступні задачі:

сформувати глосарій реалізацій ознак для кожного класу атак, а також, алфавіт класів в термінах ОР;

визначити мінімальний обсяг навчальної матриці (ОВН) (при дотриманні вимоги по її репрезентативності);

визначити нормовані допустимі відхилення для реалізацій ознак розпізнавання нелегітимного втручання в роботу КВІС.

Для того щоб отримати вхідний математичний опис АСР потрібно детально вивчити й проаналізувати особливості функціонування первинних джерел інформації, з яких система розпізнавання отримує дані про прояву певних реалізацій ознак кібератаки. Наприклад, в існуючих СВВ, в якості

первинних джерел інформації використовують дані антивірусних систем та «агентів» виявлення вторгнень [75, 80, 107, 114, 190].

Математична модель АСР у загальному вигляді як теоретико-множинна структура, може бути подана у наступному вигляді [50, 51, 79, 144]:

$$\Delta_{\text{В}} =< IS, T, RS, SS, OS, \Pi, \Phi >, \qquad (2.2)$$

де IS – множина вхідних сигналів (факторів), які опрацьовуються АСР;

T – моменти часу для отриманні інформації про стан КВІС яка підлягає захисту;

RS – множина реалізацій ознак, які використовуються в процесі розпізнавання (або ББНМ);

SS – простір можливих станів КВІС яка підлягає захисту**;**

OS – множина даних, які отримані на виході модуля первинного опрацювання сигналів (інформації) – МПОД;

$\Pi : IS \times T \times RS \to SS$ – квантор переходів (використовується для фіксації зміни станів КВІС яка підлягає захисту. Приймається, що зміна станів відбувається під впливом внутрішніх або/чи зовнішніх факторів);

$\Phi : IS \times T \times RS \times SS \to LM$ – квантор формування множини LM (learning matrix – навчальної матриці).

У якості універсуму UT випробувань під час тестування АСР використовується декартовий добуток множин IS, T, RS, SS:

$$UT = IS \times T \times RS \times SS. \qquad (2.3)$$

Схема АРС, яка включає ЕС, наведена на рис. 2.3.

Квантор $O\Theta : CT^{|2|} \to RC^{|2|}$ використовується для розбиття простору реалізацій ознак ОР (кібератак) на два класи розпізнавання.

Параметр класифікації OC використовується для перевірки статистичного припущення (тобто гіпотези) про приналежність ОР до модельованого класу кібератак.

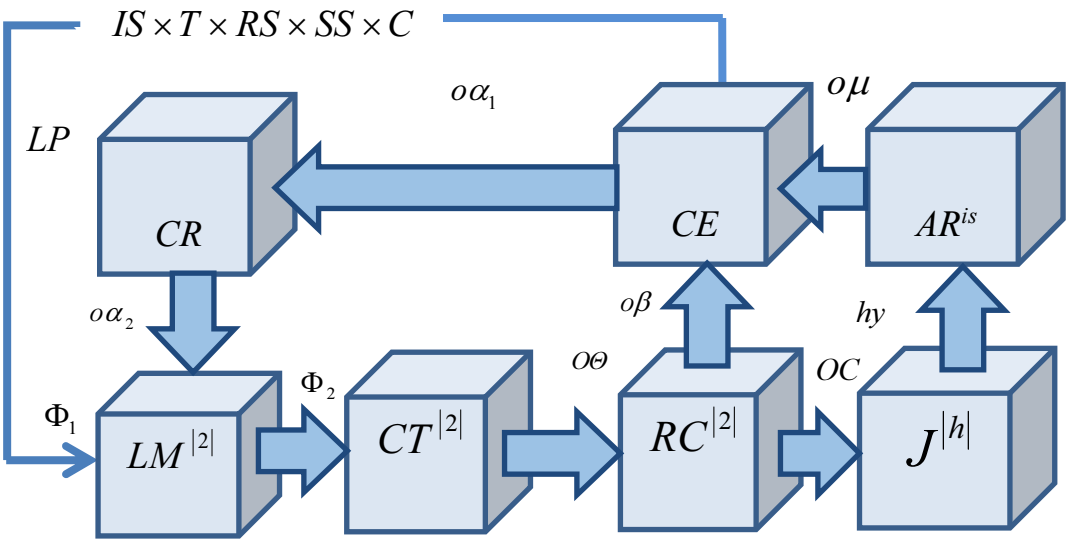


Рис. 2.3. Принципова схема АСР

Після оцінки статистичних припущень (гіпотез) за допомогою квантору hy, формується множина AR^{is} яка характеризує точність розпізнавання кібератаки у АСР та ЕС, зокрема. Прийнято h – кількість статистичних припущень, $is = h^2$ – кількість характеристик АСР.

Квантор $o\mu$ формує множину CE, яка дозволяє виконувати процедуру оцінювання ефективності розпізнавання відповідного ОР в межах класу.

Квантор $o\beta$ замикає контур розпізнавання та застосовується для оптимізації системи контрольних відхилень від шаблонів, які зберігаються у репозиторії ОР.

Квантори $\Phi_1 : IS \times T \times RS \times SS \times C \to LM^{|2|}$ та $\Phi_2 : LM^{|2|} \to CT^{|2|}$ використовуються для формування вхідної навчальної матриці (ВНМ) та організації бінарної навчальної матриці (БНМ), відповідно.

Множина CR, замикається послідовно кванторами $o\alpha_1 : CE \to CR$ та $o\alpha_2 : CR \to ML^{|2|}$, які дозволяють змінювати реалізації реалізацій ознак кібератак різних класів в процесі навчання АСР.

Для регламентування процесу навчання ЕС у складі АСР використовується квантор $LP : CE \to IS \times T \times RS \times SS \times C$.

На підставі схеми, рис. 2.3, сформулюємо наступну формалізовану постановку завдання інформаційного синтезу елементів АСР, зокрема й ЕС. Нехай відомі алфавіт класів ОР $\left\{CT_m^o \mid m = \overline{1,M}\right\}$ й ББНМ ОР, яка, відповідно, характеризує m-й функціональний стан системи для класу розпізнавання CT_m^o:

$$\left\| lm_{m,i}^{(j)} \right\| = \begin{vmatrix} lm_{m,1}^{(1)} & lm_{m,2}^{(1)} & \cdots & lm_{m,k}^{(1)} & \cdots & lm_{m,N}^{(1)} \\ lm_{m,1}^{(2)} & lm_{m,2}^{(2)} & \cdots & lm_{m,k}^{(2)} & \cdots & lm_{m,N}^{(2)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ lm_{m,1}^{(j)} & lm_{m,2}^{(j)} & \cdots & lm_{m,k}^{(j)} & \cdots & lm_{m,N}^{(j)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ lm_{m,1}^{(n)} & lm_{m,2}^{(n)} & \cdots & lm_{m,k}^{(n)} & \cdots & lm_{m,N}^{(n)} \end{vmatrix}. \qquad (2.4)$$

У виразі (2.4) прийнято наступні позначення: рядок матриці –реалізація «подання» ОР $\left\{lm_{m,i}^{(j)} \mid i = \overline{1,N}\right\}$, N – кількість інформативних реалізацій ознак розпізнавання; стовпчик – стохастична вибірка $\left\{lm_{m,i}^{(j)} \mid j = \overline{1,n}\right\}$, яка використовується під час навчання, n – обсяг вибірки.

Чітка організація глосарію реалізацій ознак розпізнавання (ГОР) – обов'язкова вимога для АСР та ЕС, зокрема. Процедура наповнення ГОР

$\sum^{|N|}$, де $N = DS\sum^{|N|}$, виконується як послідовність дій спрямованих на формалізацію реалізацій ознак. Первини ознаки безпосередньо характеризують певну атаку, а вторинні ознаки, у свою чергу, походять від первинних.

В якості первинних реалізацій ознак можна використовувати параметри які зчитуються з певних датчиків, або експериментальні дані, отримані безпосередньо, наприклад в ході реалізації тестів на проникнення у КВІС.

В якості вторинних реалізацій ознак ОР можна використати різноманітні статистичні характеристики, наприклад, вектори реалізації певного класу $\{lm_{m,i}^{(j)} \mid i = \overline{1,N}\}$, навчальну вибірку $\{lm_{m,i}^{(j)} \mid j = \overline{1,n}\}$ для ОВН та ін.

Алфавіт класів ОР для АСР $\{lm_m^o\}$ формується на першому етапі розробником системи із залученням фахівців із ІБ.

На другому етапі синтезу алфавіту продовжується за допомогою СППР або ЕС, яка безпосередньо здатна функціонувати в режимі кластер-аналізу вхідних даних, процедура його опрацювання.

Як було раніше показано у роботах [50-52, 80, 93, 148, 211] у випадку незмінності глосарію реалізацій ознак ОР та збільшенні ємності алфавіту, можлива зміна асимптотичних характеристик АСР. Відповідно, цей чинник може суттєво вплинути на функціональну ефективність процедури навчання подібних систем. Це, зокрема обумовлено, збільшенням ступеню перетину класів кібератак які підлягають розпізнаванню (далі – ОР).

На рис. 2.4 показано процес формування структури навчальної матриці, яка поетапно включає вектори реалізації $\left\{ct_1^{(j)}\right\} \in CT_1^o$ та $\left\{ct_2^{(j)}\right\} \in CT_2^o$, відповідно.

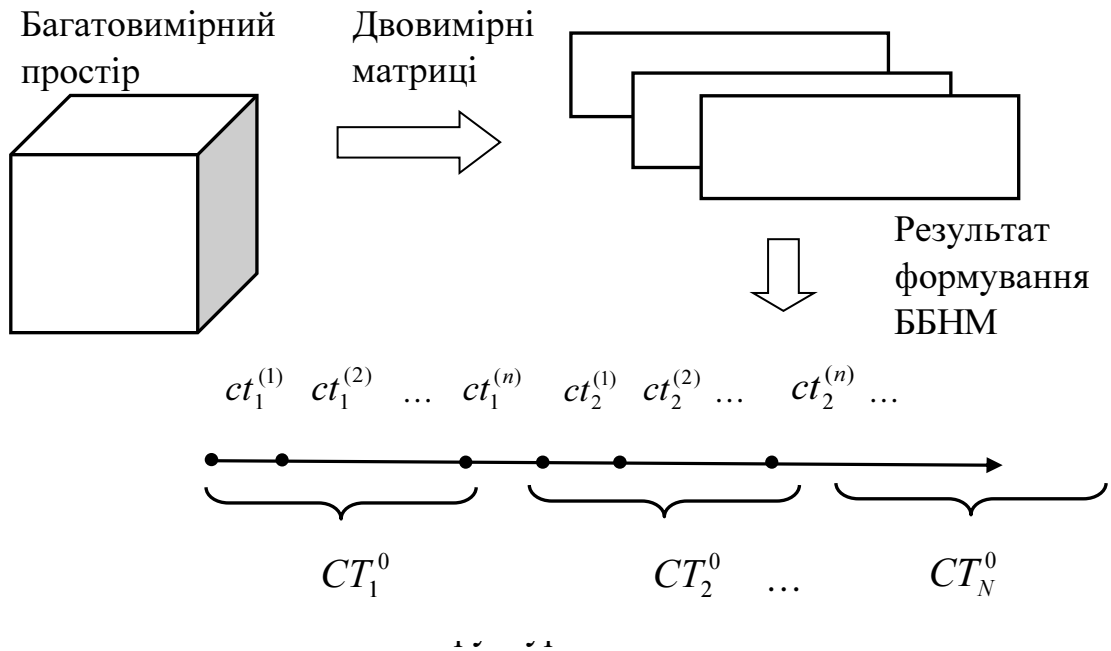


Рис. 2.4. Схема роботи з багатовимірним інформаційним простором реалізацій ознак для АСР кібератак на КВІС

Для побудови такої матриці повинні бути виділені тільки значимі властивості ОР, які однозначно відрізняють одну автоматично виявлену кібератаку в межах класу від іншої. Зрозуміло, що для кожної АСР КВІС класифікація уразливостей може бути своя. Проте, більшість з ОР містить такі властивості, як, наприклад, тип уразливості, протокол, за яким уразливість може бути використана, канал реалізації всередині цього протоколу, тип об'єкту, шлях до об'єкту та ін., див. табл. 2.1.

Всі можливі значення кожної властивості ОР можна кодувати у бінарній формі [80, 113], або за допомогою невід'ємних цілих чисел [140, 148], де нуль відповідає невизначеному значенню властивості ОР. Це, зокрема, дає змогу враховувати, відсутні, нові або поки не передбачені значення властивості ОР.

У таблицях 2.2 і 2.3 показані приклади бінарних матриць ідентифікованих джерел загроз і уразливостей для класів кіберзагроз,

пов'язаних з побічним випромінюванням елементів технічних засобів, кабельних ліній КВІС (КВКС), а також спроб кібератак.

Таблиця 2.1

Приклад формування матриці реалізацій ознак

Стовпчик / Загроза	0	1	2	…
Тип	Невідомо	SQL	XSS	…
Протокол	Невідомо	FTP	HTTP	
Канал	Невідомо	Post	get	…
Тип об’єкту	Невідомо	js	php	…
…				
Ін.	Невідомо	…		

Ознаки кібератак виявляються у великій кількості вимірюваної інформації, наприклад, логи, дані моніторингу та ін., Це, у свою чергу, вимагає підвищення швидкості опрацювання інформації у СВВ.

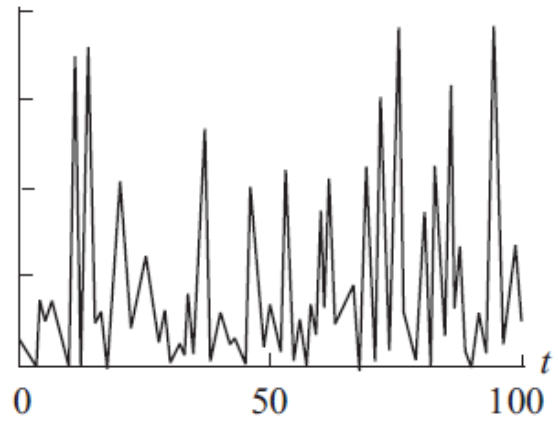
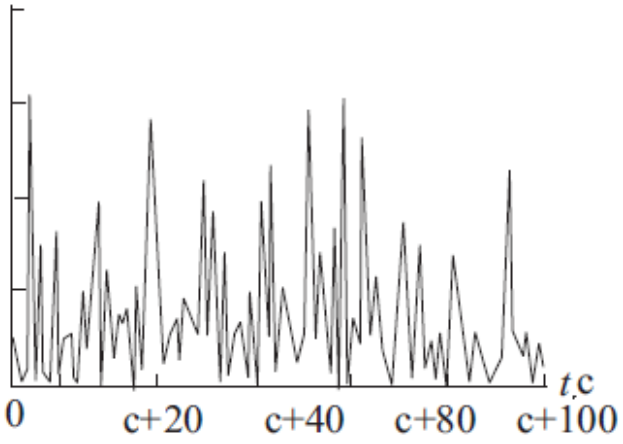
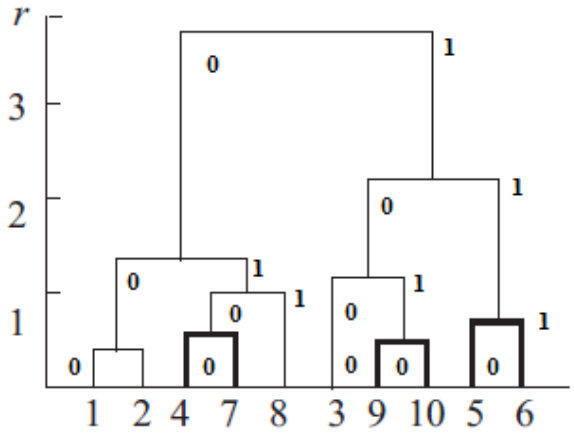
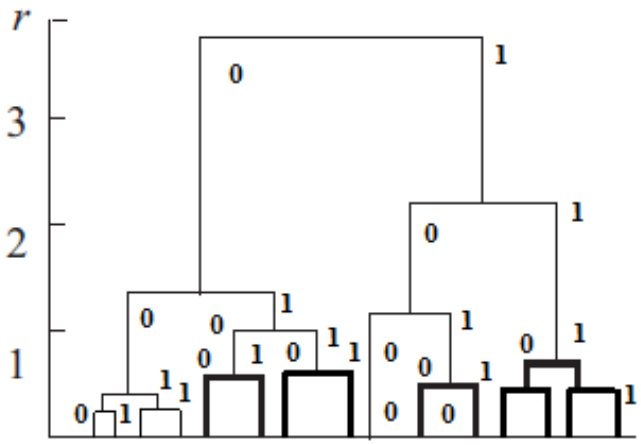
Таблиця 2.2

Матриця ідентифікованих джерел загроз для класу випромінювань

Джерела загроз / Уразливості	Хакери	Несумлінні конкуренти	Технічний персонал постачальників прикладного ПЗ	Технічний персонал постачальників комунікаційних послуг	Основний персонал	Адміністратори КВІС	Представники служб безпеки (у тому числі ІБ)	Технічний персонал служб з експлуатації мереж, серверів і т.п.	Кримінальні структури	Іноземні спецслужби
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Побічні випромінювання елементів технічних засобів КВІС (АРМ, копіювальних апаратів тощо)	0	1	0	0	0	0	1	1	1	1
Випромінювання кабельних ліній КВІС	0	1	0	0	0	0	1	1	1	1
…	…	…	…	…	…	…	…	…	…	…

Таблиця 2.3

Приклад формування кластерів та БНМ для ідентифікованих класів мережевих кібератак на КВІС (КВКС)

Атака відсутня	**Зафіксована атака**
Структурній вигляд трафіка	
	
Формування кластерів	
	
Формування БНМ	
$lm = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$	$lm = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & - & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & - & 0 & 0 \end{pmatrix}.$

Об'єднуючи дані в компактні кластери, можна проводити аналіз типових представників кожного кластера й приймати рішення про те, чи є такі дані ознакою атаки чи ні. Потім це рішення переноситься на всіх представників досліджуваного кластера. Такий підхід суттєво скорочує обсяги необхідної для успішної класифікації атаки інформації (ОВН).

Оскільки кластери можуть приймати в багатовимірному просторів реалізацій ознак складні форми, деякими авторами запропоновані різні алгоритми кластеризації реалізацій ознак, наприклад, K–means [132, 148, 151] DBSCAN [132, 188, 190], FDBSCAN [164, 188] та ін., див. рис. 2.5.

Обчислювальна трудомісткість алгоритмів які використовуються у бінарному просторі реалізацій ознак розпізнавання RS_b (БПОР), відповідного класу (класів), залежить від оптимальної форми контейнеру для відповідного класу об’єкту розпізнавання.

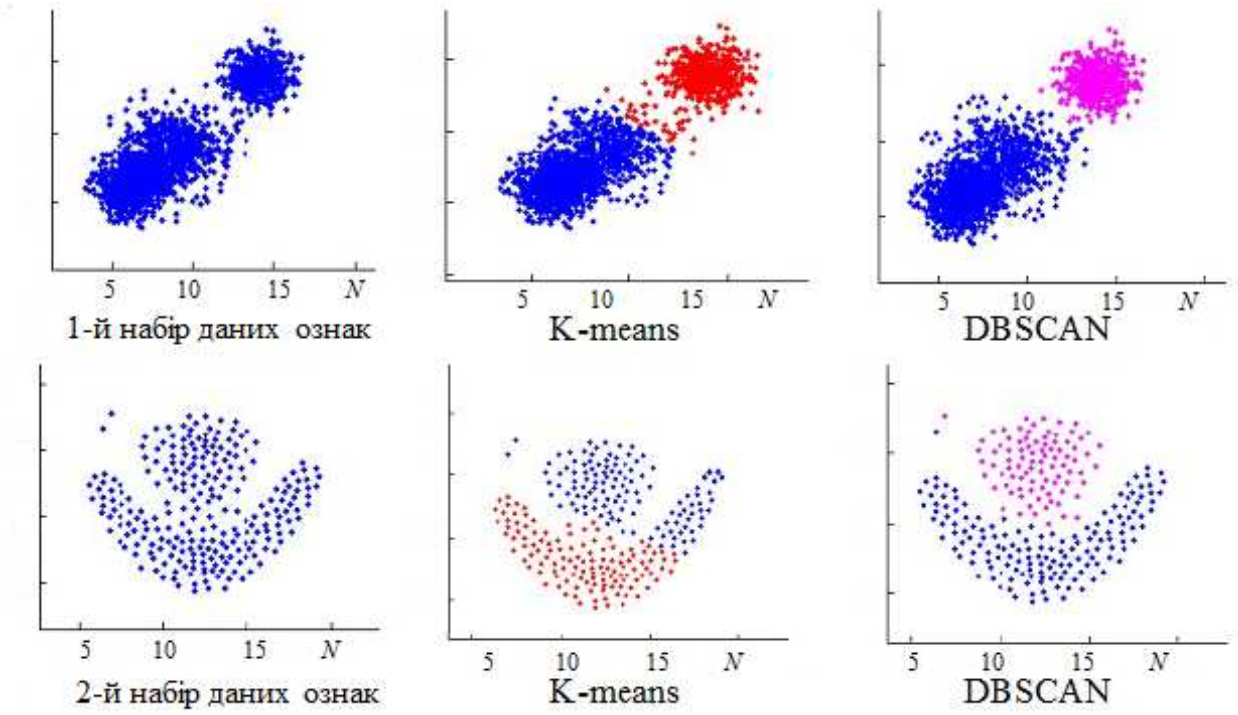


Рис. 2.5. Приклади кластеризації даних алгоритмом DBSCAN та K–means у СВВ

Для більш зручної побудови контейнеру зробимо наступне припущення: існує контейнер – КОН [50–52, 80, 146], який допускає розгляд параметрів оптимізації КОН в БПРО (RS_b), як деякого вектору–еталону, наприклад, $ct_m \in CT_m^o$. Вершина вектору визначить геометричний центр КОН – C_m^o, для розрахунку радіусу (container radius) якого скористаємось виразом [50, 51, 79, 132, 151, 164, 188]:

$$cr_m = \sum_{i=1}^{N} \left(ct_{m,i} \oplus \zeta_i \right),$$

де $ct_{m,i}$ – i-та координата вектору-еталону ct_m;

ζ_i – i-та координата вектору ζ реалізації ОР, вершина якого відноситься до КОН $C_m^o \in CT_m^o$

N – кількість реалізацій ознак ОР.

Алгоритм K-means має низьку обчислювальну складність, це основна перевага K-means і він добре працює з великим числом даних. DBSCAN досить повільно працює з великою кількістю даних.

Припустимо, що проведена серія вимірювань значень контрольованих реалізацій ознак в КВІС, і отримана матриця наступного вигляду:

$$S = \begin{pmatrix} 0 & 1 & \dots & 1 & \dots & 1 \\ 1 & 0 & \dots & - & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ - & 1 & \dots & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & - & \dots & 0 \end{pmatrix}.$$

Таким чином, набір об'єктів, що перевіряються, що належать класу, задається бінарними ознаками {1001…–01}. Прочерк вказує на невизначеність, ознаки у ОВН.

Більш детальні результати досліджень процедур формування БПОР та бінарних навчальних матриць (ОВН), наведено у роботах [141, 159, 165, 197] а також патентах США: US 2011/0208714 A1; US9294502 та ін.

Будемо вважати, що для оцінювання функціональної ефективності здатної навчатися АСР, потрібно враховувати вплив на параметри її роботи структурованого вектору просторово-часових параметрів – $is =< is_1, \ldots, is_k, \ldots, is_{RS} >$, з відповідними обмеженнями $RC_k\left(is_1, \ldots, is_{RS}\right) \le 0$. Використовуючи ІЕТ навчання, за мету ставиться визначення оптимальних значень параметрів реалізації вектору $\left\{is_c^*\right\}$ в області визначення його функції, які містять $\max$ ІУФР [26, 50–52, 62, 194]:

$$CE_m^* = \max_{IS} CE_m, \tag{2.5}$$

де CE_m – ІУФР процедури машинного навчання АСР в ході розпізнавання класу ОР CT_m^0;

IS – допустимі значення параметрів функціонування АСР.

Таким чином, в ході частинного інформаційного синтезу АСР (або ЕС з ІБ) частково розв’язати задачу можна визначивши оптимальне значення параметра is_k^*:

$$is_k^* = \arg_{IS_k} CE_m^*, \tag{2.6}$$

де IS_k – область допустимих значень параметра is_k.

У таблиці 2.4 наведено перелік основних джерел даних для АСР та інформація, що підлягає опрацюванню та аналізу, наприклад, за допомогою ЕС.

На рис. 2.6 наведено функціональну схему АСР, до складу якої входить здатна до самонавчання ЕС з питань ІБ КВІС (КВКС). Для наочності на схемі показані основні функціональні вузли та інформаційні потоки, зокрема, фігурними стрілками показані зв'язки між функціональними модулями ЕС (див. табл. 2.5) у складі АСР, а звичайними стрілками зображені керуючи команди.

Під час навчання АСР та формування БЗ роботу системи регламентує фахівець з ІБ, який згідно з рекомендаціями ЕС формує керуючі команди (control commands – КК) – $\left\{CC\left\{hy_m\right\} \mid m=\overline{1,M}\right\}$.

МФБВР (див. табл. 2.5) здійснює бінеарізацію векторів-реалізацій класів ОР шляхом порівняння поточних реалізацій ознак з їх відповідними перевірочними допустимими відхиленнями $\left\{ca_{K,i}\right\}$, які містяться у БД. Залежно від заданого режиму МФБВР у складі АСР розробляємої ЕС, формує багатовимірний двійковий вектор (БДВ) який є параметром-реалізацією подання ОР в ЕС. Кожну координату БДВ при алгоритмічній реалізації АСР (або ЕС) можна подати як одномісний предикат, який дорівнює "1", для випадку – значення інформативної ознаки ОР належить діапазону перевірочних допустимих відхилень (ДПДВ), та дорівнює "0", якщо – не належить.

Таблиця 2.4

Перелік основних джерел даних для АСР

Джерело даних для побудови кластерів	**Інформація яка підлягає опрацюванню та аналізу**
log-файли працюючих підсистем КВІС (КВКС)	Час і тип виконуваних операцій, сутність операцій, відповідність пароля, збій при
Мережевий трафік	Завантаження мережевого обладнання, використання каналів зв'язку, мережева
Довідники і журнали реєстрації користувачів і	ID-коди користувачів, коректність паролів, виконувані дії
Перелік функціональних завдань	Ланцюжки взаємопов'язаних викликів завдань і процесів
Інформація про права	Дотримання регламенту звернень до ресурсів
Відомості про роботу поштової системи	Статистика, обсяги і адресність посилок і поштових надходжень, тематика повідомлень
Текстові файли	Тематична спрямованість
Прикладне ПЗ	Попередня процедура аудиту ІБ
Таблиці з атрибутами виконуваних файлів	типи файлів, дати створення і зміни, автори змін і їх права, контроль «незмінності», адреси еталонних модулів, контрольні суми
Інше	

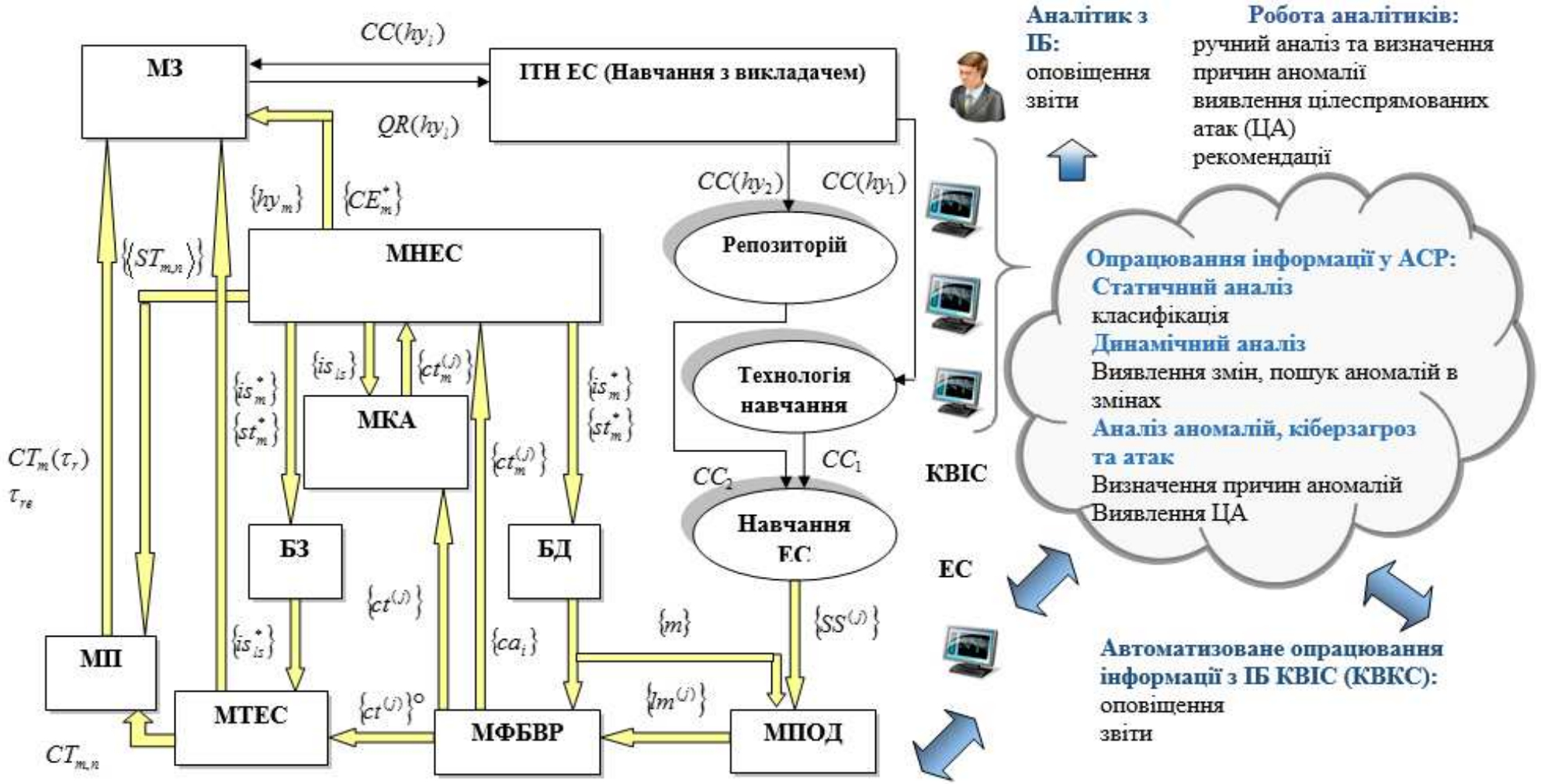


Рис. 2.6. Функціональна схема АСР

Таблиця 2.5

Перелік основних модулів ЕС «Analyzer of cyberthreats» у складі адаптивної системи розпізнавання

№	Позначення	Опис	Математичний апарат	Джерела
1	БД	База даних ЕС	Теорія реляційних БД	[60, 84, 87]
2	БЗ	База знань ЕС	Моделі на основі нечітко-множинного підходу	[52, 62, 70]
3	МЗЕС	Модуль запитів до ЕС	Онтологічні моделі та апарат алгебраїчної теорії типів	[73, 84, 105]
4	МКА	Модуль кластерного аналізу	Модель системи кластеризації з використанням алгоритму нечіткої кластеризації fuzzy C-means та застосуванням критерію оцінки якості попереднього опрацювання даних критерія ентропії Шеннона і відносної зміни ентропії у процесі трансформації даних та Модель системи кластеризації з використанням інформаційного критерія Кульбака – Лейблера	[26, 80, 94, 132, 148, 151, 164, 188, 190]
5	МКОД	Модуль кінцевого опрацювання даних	Теорія управління системами, моделі опрацювання статистичних даних	[5, 112, 148]
6	МНЕС	Модуль навчання ЕС	Теорія та методи штучного інтелекту, моделі машинного навчання, MIET	[94, 80, 114]
7	МП	Модуль прогнозування	Моделі оцінювання (cybersecurity situation assessment, CSSAs) та прогнозування (cybersecurity situation forecast, CSSF) кібербезпеки об’єкту Модель поширення комп'ютерних вірусів - PSIDR (Progressive Suspected-Infected-Detected-Recovered).	[150, 161]
8	МПОД	Модуль попереднього опрацювання даних	Теорія управління системами, моделі опрацювання статистичних даних	[5, 72, 164]
9	МТЕС	Модуль тестування ЕС	Математичні моделі діагностичного контролю ЕС в процесі їх експлуатації	[33, 46, 51]
10	МФБВР	Модуль формування бінарних векторів розпізнавання	Булева алгебра, дискретна математика	[21, 26, 28, 141, 159, 165, 197]

Розглянемо процедуру функціонування ЕС як елемента АСР (СІРКЗ) у режимі підготовки за апріорно класифікованою навчальною матрицею (КНМ) (навчання з “учителем”).

При впливі на керований процес підготовки ЕС стохастичних факторів $rf(t)$ і довільних початкових умов формування реалізацій $\left\{ss^{(j)},\ j=\overline{1,n}\right\}$ функціонального стану КВІС в умовах реалізації кібератаки, в МПОД здійснюється, згідно з вибраною оціночною функцією, формування шкали розцінювання й відображення на неї поточної реалізації $ss^{(j)}$ з метою отримати реалізацію $lm_m^{(j)}$, координати якої є нормованими результатами спостереження за станом КБ. Крім того, МПОД здійснює перевірку статистичної стійкості й однорідності навчальних вибірок за відповідними статистичними критеріями [5, 26, 80, 94, 164, 190] та визначення мінімального обсягу $n_{\min}$ навчальної вибірки за умови її репрезентативності, наприклад, за методами які містяться у таблиці 2.4, тобто виконує функції попереднього аналізу. На виході МПОД, залежно від заданої ємності m алфавіту класів розпізнавання, АСР створює класифіковану нечітку навчальну матрицю $\left\{lm_m^{(j)}\right\}$, яка подається на вхід МФБВР.

У результаті у МФБВР створюється бінарна навчальна матриця (БНМ) $\left\{ct_m^{(j)} \mid j=\overline{1,M}\right\}$, яка складається із структурованих стохастичних векторів-реалізацій образу відповідної кібератаки:

$$ct_m^{(j)} = \left\langle ct_{m,1}^{(j)},\ldots,ct_{m,i}^{(j)},\ldots,ct_{m,N}^{(j)} \right\rangle, \qquad (2.7)$$

де N – кількість інформативних реалізацій ознак розпізнавання.

БНМ в подальшому також використовується для оцінки перевірочних допустимих відхилень в процесі розпізнавання (система перевірочних/контрольних допустимих відхилень – СКДВ). СКДВ

$\{ca_{n,i} \mid i = \overline{1,N}\}$, а також параметри, які визначають рівні вибірки $\{cl_m\}$ координат бінарних еталонних векторів класів ОР, поступають в МФБВР із бази даних.

В режимі навчання ЕС на виході МФВР за час τ_{ed}, формується ББНМ – $\{ct_m^{(j)} \mid m = \overline{1,M}; j = \overline{1,n_{\min}}\}$, яка поступає на вхід модуля «Навчання ЕС» (МНЕС). Зазначимо, що формування ББНМ відбувається за деяким наперед заданим рівнем довіри [5, 132, 188].

З виходу МНЕС у БЗ надходить вектор оптимальних параметрів (ВОП) функціонування ЕС:

$$\{is_k^*\}^{\mathrm{O}} \mid k = \overline{1,\mathrm{O}}; \mathrm{O} = \overline{1,\Omega}\}, \tag{2.8}$$

де O – відображення відкритості множини, або у випадку реалізації процедури розпізнавання – число реалізацій ОР).

ВОП забезпечує $\max$ значення ІУФР в допустимій області його визначення. Коли метою навчання ЕС є складення прогнозних сценаріїв для оцінювання можливих змін станів КВІС у результаті впливу кібератак, крім цих параметрів, обчислюються значення одновимірних екстремальних статистик $\{ST_m^*\}$. Під час тестування ЕС, а саме в момент $\tau_{EC_{\max}}$, коли ІУФР $EC_m^* = \max$, процес навчання для розпізнавання реалізації класу CT_m^o зупиняється. Також для $\tau_{EC_{\max}}$ поточні статистичні параметри ST_m, які є членами відповідного варіаційного ряду, приймаються як екстремум функціонального розподілу ST_m^*.

В режимі тестування ЕС, тобто безпосереднього прийняття рішень які дозволяють здійснювати розпізнавання складних кібератак, із МФБВР в модуль «Тест» (МТЕС) надходить тестова матриця $\{ct^{(j)}\}^{\mathrm{O}}$. Одночасно в

МФБВР із БЗ надходять оптимальні значення перевірочних допустимих відхилень $\left\{ca^{*}_{K,i}\right\}$ і рівнів вибірки $\left\{sl^{(j)}_{m}\right\}$ для бінарних еталонних векторів класів ОР. Це дозволяє гарантовано забезпечити еквівалентні умови формування навчальних та екзаменаційних матриць.

З першого виходу МТЕС аналітик з ІБ КВІС через «Модуль запитів» (МЗЕС) має можливість отримувати підозри hy_m про належність до можливого стану АСР та КВІС класу CT^{o}_{m} та, відповідно, розробляти адекватні заходи реагування на виниклу кібератаку – траєкторія реагування (ТР).

В режимі кластерного аналізу даних, які надходять у АСР, та для розв'язання завдання автоматизації процедури формування вхідної класифікованої навчальної матриці (ВКНМ або ОВН) із МФБВР на перший вхід МКА (модуль «Кластерний аналіз» – МКА) ЕС, подається некласифікована навчальна матриця (ННМ) – $\left\{ct^{(j)}\right\}$. ННМ складається з реалізацій всіх класів ОР та відповідного алфавіту.

Класифіковані навчальні матриці (КНМ) – $\left\{ct^{o}_{m}\right\}$, сформовані на кожному кроці кластеризації вхідних даних у АСР, подаються на другий вхід модуля «Навчання ЕС» (МНЕС). Відповідно, цей модуль відповідає за процес оцінювання за ІУФР якості виконаної процедури кластеризації та видає на другий вхід МКА значення параметрів кластеризації $\left\{is_k\right\}$. Отримані параметри $\left\{is_k\right\}$, далі оптимізуються за заздалегідь визначеним алгоритмом [26, 80, 94, 151, 164, 188, 190].

Таким чином, алгоритм кластерного аналізу вхідних даних в ЕС та МІЕТ є частиною алгоритму навчання АСР кібератак.

Важливою властивістю ЕС з ІБ КВІС є можливість прогнозувати зміну її функціональної ефективності в процесі розпізнавання кібератак, а також, та визначення моменту коли з'являється потреба у перенавчанні системи, наприклад, у випадках появи нових раніше не класифікованих типів загроз та

кібератак. У цьому випадку на перший вхід модуля прогнозування (МП) надходять поточні статистичні дані $ST_{m,n}$, які опрацьовуються модулем «Тестування ЕС» (МТЕС). Ці дані характеризують статистичні властивості бінарної екзаменаційної матриці класу CT_m^o, який визначається відповідними вирішальними правилами, отриманими під час навчання ЕС.

На другий вхід МП з БЗ поступає статистичний масив $\{< ST_m^* >\}$, який характеризує відповідні статистичні властивості класів ОР в момент першого навчання τ_1 ЕС та має властивість інваріантності до законів розподілу ймовірностей. Точність й достовірність прогнозування напряму залежить від отриманого в процесі навчання ЕС в момент прогнозування τ_r значення критерію $CE_m^*(\tau_r)$.

За результатом роботи МП (Модуль прогнозування) отримуємо параметричний клас розпізнавання $-\{CT_m^o(\tau_r)\}$. Цей клас характеризує функціональний стан АСР в момент часу τ_r, і у випадку відхилення статистичних даних $ST_{m,n}$ за межі варіаційних параметрів які містяться у модулі прогнозування та опрацювання статистичних даних (МП) – ST_m^* приймається рішення про необхідність перенавчання ЕС як елемента АСР. Таким чином, на підґрунті отриманих результатів у МП в момент часу τ_{re} приймається рішення про необхідність перенавчання ЕС як елемента АСР.

Розглянута структура ЕС відрізняється від існуючих широкими функціональними можливостями і дозволяє розв’язувати складні задачі забезпечення надійного кіберзахисту КС, як при вже сформованій БЗ для відомих класів ОР, так і під час її машинного навчання, у разі появи нових раніше невідомих класів кібератак.

2.2. Інформаційні критерії функціональної ефективності навчання адаптивної системи розпізнавання

В процесі розробки АСР неодмінно постає питання оцінювання функціональної ефективності процесу машинного навчання. Зокрема, це дозволяє визначити максимальну асимптотичну достовірність рішень, які приймаються під час тестування АСР в ході виявлення окремих класів кібератак. Для технології [26, 50–52, 80, 94, 149] навчання АСР, зокрема, й ЕС можливо використовувати різні критерії, які задовольняють наступним властивостям інформаційних мір (ІМ) [9]: ІМ – дійсний параметр [5, 9]; для детермінованих змінних ($p_i = 1$ або $p_i = 0$) кількість інформації дорівнює нулю; ІМ має екстремум при ймовірності $p_i = \frac{1}{m}$, де m – чисельність інформативних реалізацій ознак розпізнавання.

Для АСР в якості інформаційних мір можна застосувати ентропійну міру [5, 7, 11, 21] та критерій Кульбака – Лейблера [5, 9, 18, 26, 50-52].

Ентропію КВІС можна розглядати як кількість інформації, пов'язаної зі структурою системи та її станами [128]. Тобто, саме ентропія дозволяє судити про технічний стан КВІС, що працює у звичайних умовах або під час кібератак. Таким чином, ентропія може розглядатися як міра «структурованості» деякого стану SS_i або міра віддаленості структури одного стану від іншого.

Тоді стохастичний процес (СП) у КВІС, що характеризує стан систем та функціонує в інтервалі часу від τ_0 до Т, описується вектором змінних стану ІБ:

$$SS_i(\tau) = f(SX(\tau), he) + hl(\tau), \tag{2.9}$$

де $he, hl(\tau)$ – «шуми» загальної природи;

$SX(\tau)$ – вектор змінних станів системи, наприклад, як наслідок кібератаки або реалізації іншої загрози для КВІС.

Спостереження величини $SS_i(\tau)$ здійснюється в моменти часу $\tau_i = \tau_0 + j\Delta, \quad j = \overline{0,n},$ із кроком дискретизації $\Delta > 0$.

Поставимо у відповідність кожному виділеному стану системи (для альтернативних припущень $hy = \{hy_1, ...hy_m\},$ що становлять повну групу подій і фізично інтерпретують стани системи) кластер [132, 151, 190]:

$$M\Theta_i = \{l_{M\Theta_i}(ss) \cdot ss \mid ss \in UK_{sig}, l_{M\Theta_i}(ss) \in ZR\}, \qquad (2.10)$$

де $l_{M\Theta_i}(ss)$ – функція числа екземплярів кластеру, що визначає кратність елемента системи $ss \in UK_{sig}$;

UK_{sig} – множина, потужність якого дорівнює максимальному рівню сигналу, характерного для ознаки об'єкта.

З урахуванням положень, наведених у розділі 2.1, узагальнимо основні етапи процедури розпізнавання у АСР:

1. Визначаємо характеристичні ознаки для кожного ОР.

2. Складаємо для кожного вузла КВІС повну групу станів системи – $hy = \{hy_1, ...hy_m\},$ яким будуть відповідати первинні специфікації $M\Theta_i$.

3. Визначаємо оцінки розподілу ймовірностей P_{SS_i} характерних для станів системи, у які вона потрапила внаслідок кібератаки.

4. Розраховуємо зміну ентропії усіх підсистем КВІС H_{SS^*} за формулою:

$$H_{SS^*} = -\sum_{i=1}^{\max} P_{SS_i} \cdot \log_2 P_{SS_i}. \qquad (2.11)$$

5. Формуємо за результатами спостережень $(SS^*_L = \{SS^*(\tau\tau), S^*(\tau+1), \ldots, SS^*(\tau+L-1)\}$ відповідний кластер:

$$M\Theta^*_L = \{is_1^L, is_2^L, \ldots, is_M^L\}, \qquad (2.12)$$

де is_j^L – зустрічаємість сигналів, характерних для j-го стану системи;

L – контрольне «вікно» [7, 148, 164].

6. Обчислюємо інформаційні відстані між кластерами $DIS(M\Theta_i, M\Theta^*_L)$ $(i = \overline{0, I})$ за $RS \geq 1$ ознаками відмінності.

7. Приймаємо рішення на користь стану, для якого величина $DIS(M\Theta_i, M\Theta^*_L)$ є найменшою для кожної ознаки RS_i. Одночасно обчислюємо вагові коефіцієнти окремих рішень:

$$kf_1^j = arg \min_{i=\overline{0,I}} DIS(M\Theta_i, M\Theta^*_L),$$
$$kf_2^j = arg \min_{i=0, I, i\neq i_1^j} DIS(M\Theta_i, M\Theta^*_L), (j = \overline{1, J}). \qquad (2.13)$$

8. Обираємо відповідно до процедури голосування [5, 9, 28, 151] той стан системи, для якого ваговий коефіцієнт більше:

$$kf_1 = \arg\min_{i=\overline{0,I}} kf_1^j, \quad kf_2 = \arg \min_{i=\overline{0,I}, i\neq i_1^j} kf_2^j. \qquad (2.14)$$

Величину нормованої ентропійної ІУФР ($CE_m^{(ls)}$), яка визначає результативність навчання ЕС під час процедури розпізнавання реалізації класу CT_m^o, попередньо представимо у наступному вигляді [9, 151, 162]:

$$CE_m^{(ls)} = \frac{I_m^{(ls)}}{I_{\max}^{(ls)}} = \frac{H_m - H_m^{(ls)}(hy)}{H_m}, \tag{2.15}$$

де $I_m^{(ls)}$ – кількість умовної інформації, що опрацьовується на $ls-му$ кроці навчанні ЕС у складі АСР для розпізнавання реалізації класу CT_m^o;

$I_{\max}^{(ls)}$ – можлива max кількість умовної інформації, яка була одержана для $ls-го$ кроку навчання ЕС у складі АСР;

H_m – апріорна (безумовна) ентропія, що існує для $ls-го$ кроку навчання ЕС у складі АСР виявляти реалізацію ОР, який належить класу CT_m^o,

$$H_m = -\sum_{m=1}^{M} p(hy_m)\log_2 p(hy_m); \tag{2.16}$$

$H_m^{(k)}(hy)$ – умовна (апостеріорна) ентропія, що встановлює невизначеність яка зберіглася на $ls-му$ кроці навчання АРС під час виявлення класу ОР CT_m^o;

$$H_m^{(ls)}(hy_l) = -\sum_{l=1}^{M} p(hy_l)\cdot \cdot\sum_{m=1}^{M} p(hy_m/hy_l)\log_2 p(hy_m/hy_l), \tag{2.17}$$

де $p(hy_l)$ – апріорна ймовірність прийняття припущення (гіпотеза) hy_l;

$p(hy_m/hy_l)$ – апостеріорна ймовірність прийняття припущення hy_m за умови, що було обрано варіант hy_l;

M – кількість розглянутих припущень в процесі розпізнавання.

На практиці під час застосування машинного навчання СІРКЗ, приймають наступне допущення – рішення має дві альтернативи, тобто $M = 2$. Оскільки АСР та ЕС, зокрема, функціонують за умов апріорної невизначеності, то згідно з принципом Бернуллі-Лапласа [5, 26], апріорна ймовірність прийняття гіпотези визначається так: $p(hy_1) = \cdots = p(hy_m) = \cdots = 1/M$.

У цьому випадку, враховуючи вирази (2.16) та (2.17) величину нормованої ентропійної ІУФР навчання (2.15) можна подати наступним чином:

$$CE = 1 + \frac{\sum_{l=1}^{2}\sum_{m=1}^{2} p(hy_m/hy_l)\log_2 p(hy_m/hy_l)}{2}. \qquad (2.18)$$

Отриманий вираз (2.18) при прийнятих допущеннях співпадає з формулою для розрахунку кількості середньої умовної інформації.

Якщо скористатися формулою Байєса для заміни апостеріорних ймовірностей на апріорні [5, 147]

$$p(hy_{\mu_m}/hy_{\gamma_l}) = \frac{p(hy_{\mu_m})p(hy_{\gamma_l}/hy_{\mu_m})}{p(hy_{\mu_1})p(hy_{\gamma_l}/hy_{\mu_1}) + p(hy_{\mu_2})p(hy_{\gamma_1}/hy_{\mu_2})}$$

та прийняти за основу концепцію двох альтернатив для ЕС ($M = 2$), а також, рівноймовірності обраних припущень (ймовірних гіпотез), які у статистичному сенсі характеризують найбільш важкі випадки прийняття рішень при аналізі ОР, то після підстановки виразів (2.16) і (2.17) в формулу (2.18), отримаємо наступний вираз для нормованої ентропійної ІУФР навчання АСР:

$$CE_m^{(ls)} = 1 + \frac{\left(\begin{array}{l} HM1 \cdot \log_2 HM1 + HM2 \cdot \log_2 HM2 + HM3 \cdot \log_2 HM3 + \\ + HM4 \cdot \log_2 HM4 + HM5 \cdot \log_2 HM5 + HM6 \cdot \log_2 HM6 \end{array} \right)}{2}, \qquad (2.19)$$

де $HM1 = mis1_m^{(ls)}(cr) / \left(mis1_m^{(ls)}(cr) + AU_{2,m}^{(ls)}(cr) \right)$;

$HM2 = mis2_m^{(ls)}(cr) / \left(AU_{1,m}^{(ls)}(cr) + mis2_m^{(ls)}(cr) \right)$;

$HM3 = AU_{1,m}(cr) / \left(AU_{1,m}^{(ls)}(cr) + mis2_m^{(ls)}(cr) \right)$;

$HM4 = AU_{2,m}^{(ls)}(cr) / \left(mis1_m^{(ls)}(cr) + AU_{2,m}^{(ls)}(cr) \right)$;

$HM5 = mis1_m^{(ls)}(cr) / \left(mis1_m^{(ls)}(cr) + AU_{3,m}^{(ls)}(cr) \right)$;

$HM6 = AU_{3,m}^{(ls)}(cr) / \left(mis3_m^{(ls)}(cr) + AU_{3,m}^{(ls)}(cr) \right)$;

$AU_{1,m}^{(ls)}(cr)$ – процедура першої валідації;

$AU_{2,m}^{(ls)}(cr)$ – процедура другої валідації;

$mis1_m^{(ls)}(cr)$ – помилки першого роду прийняття рішення для $ls-го$ кроку навчання ЕС;

$mis2_m^{(ls)}(cr)$ – помилки другого роду прийняття рішення для $ls-го$ кроку навчання ЕС;

$mis3_m^{(ls)}(cr)$– помилки під час завдання на прийняття рішення в ході процедур МНав ЕС;

cr – радіус КОН [132, 151, 164].

Визначення ентропійної ІУФР лише на основі радіального базису простору Хеммінга та функціональної залежності відстані реалізацій ОР у АСР від вершин еталонних векторів має певні недоліки [188], зокрема у випадках необхідності опрацювання великої кількості даних. Тому $CE_m^{(ls)}$ у МІЕТ можна представити як нелінійну й взаємно-неоднозначну функціональну залежність функціональної результативності АСР від характеристик ОР. Ці недоліки можна усунути шляхом трансформації інформаційної міри Кульбака – Лейблера (ІМКЛ). Взагалі ІМКЛ являє собою, добуток відношення правдоподібності параметра О (вираз 2.8) на міру відхилень відповідних розподілів ймовірностей $p\left(hy_{\gamma_{1,k}} / hy_{\mu_m}\right)$.

Раніше в роботах [26, 50-52, 80, 94, 132, 164] було розглянуто модель та методику отримання логарифмічного відношення повної ймовірності $P_{t,m}^{(ls)}$ належності реалізацій класів ОР, відповідно, CT_m^o та CT_c^o КОН $C_{m,k}^o \in CT_m^o$. Це дозволяє підвищити коректність результатів перевірки, а також, визначити ймовірність помилкового рішення $P_{f,m}^{(ls)}$.

Для системи яка дозволяє порівняти два альтернативних припущення, маємо наступний вигляд для параметра О:

$$\mathrm{O} = \log_2 \frac{p\left(hy_{\mu_m}\right)\cdot p\left(\dfrac{hy_{\gamma_{1,ls}}}{hy_{\mu_m}}\right) + p\left(hy_{\mu_c}\right)\cdot p\left(\dfrac{hy_{\gamma_{2,ls}}}{hy_{\mu_c}}\right)}{p\left(hy_{\mu_m}\right)\cdot p\left(\dfrac{hy_{\gamma_{2,ls}}}{hy_{\mu_m}}\right) + p\left(hy_{\mu_c}\right)\cdot p\left(\dfrac{hy_{\gamma_{1,ls}}}{hy_{\mu_c}}\right)}, \qquad (2.20)$$

де $p\left(hy_{\mu_m}\right)$ – безумовна ймовірність появи реалізації класу CT_m^o;

$p\left(hy_{\mu_m}\right)$ – безумовна ймовірність появи реалізації сусіднього (найближчого) класу ОР CT_c^o;

$hy_{\gamma_{1,ls}}$ – припущення про приналежність КОН $C^{o}_{m,k} \in CT^{o}_{m}$ до реалізації класу ОР CT^{o}_{m};

$hy_{\gamma_{2,ls}}$ – альтернативне припущення (гіпотеза).

Враховуючи отриманий вираз (2.20) та відповідні припущення згідно із принципом Лапласа-Бернуллі [5, 9], що $p(hy_m) = p(hy_c) = 0{,}5$, отримаємо наступний вигляд ІУФР, яка містить загальну міру Кульбака – Лейблера:

$$
\begin{aligned}
CE^{(ls)}_{Km} &= \log_2 \frac{P^{(ls)}_{t,m}}{P^{(ls)}_{f,m}} \cdot \left[P^{(ls)}_{t,m} - P^{(ls)}_{f,m}\right] = \\
&= \left| \begin{array}{l} P^{(ls)}_{t,m} = \left(AU^{(ls)}_{1,m}(cr)/2\right) + \left(AU^{(ls)}_{2,m}(cr)/2\right) \\ P^{(ls)}_{f,m} = \left(mis1^{(ls)}_{m}(cr)/2\right) + \left(mis2^{(ls)}_{m}(cr)/2\right) \end{array} \right| = \\
&= 0{,}5 \cdot \log_2 \left(\frac{AU^{(ls)}_{1,m}(cr) + AU^{(ls)}_{2,m}(cr)}{mis1^{(ls)}_{m}(cr) + mis2^{(ls)}_{m}(cr)} \right) \cdot \left[\begin{array}{l} \left(AU^{(ls)}_{1,m}(cr) + AU^{(ls)}_{2,m}(cr)\right) - \\ \left(mis1^{(ls)}_{m}(cr) + mis2^{(ls)}_{m}(cr)\right) \end{array} \right] = \\
&= \log_2 \left(\frac{2 - \left(mis1^{(ls)}_{m}(cr) + mis2^{(ls)}_{m}(cr)\right)}{mis1^{(ls)}_{m}(cr) + mis2^{(ls)}_{m}(cr)} \right) \cdot \left[1 - \left(mis1^{(ls)}_{m}(cr) + mis2^{(ls)}_{m}(cr)\right)\right]
\end{aligned}
\qquad (2.21)
$$

Таким чином, модифіковану ІУФР навчання ЕС подамо так:

$$
CE^{(ls)}_{K,m} = CE^{(ls)}_{Km} \big/ CE^{(ls)}_{K\max}, \qquad (2.22)
$$

де $CE^{(ls)}_{K\max}$ – значення ІУФР навчання ЕС при підстановці $AU^{(ls)}_{1,m}(cr) = AU^{(ls)}_{2,m}(cr) = 1$ та $mis1^{(ls)}_{m}(cr) = mis2^{(ls)}_{m}(cr) = 0$ у виразі (2.18).

Реалізація процедури нормування критеріїв оптимізації є доцільною у випадку коли необхідно проводити порівняльний аналіз результатів досліджень різних АСР, а також, оцінити ступінь близькості показників СВВ

які вже знаходяться в експлуатації та потенційно перспективних для впровадження у СІБ КВІС. Але зважаючи на складність процедури машинного навчання у АСР в завданнях забезпечення кібербезпеки КВІС, для оптимізації функціональних параметрів модулів ЕС здатної до самонавчання, на нашу думку, на попередніх етапах, можна використати більш простий параметр – ненормовану ІУФР.

Відповідно до вище наведених у розділі виразів для обчислення модифікованої ентропійної ІУФР навчання АСР (2.19 та 2.21), розглянемо процедуру прийняття рішення у АСР для двох альтернатив та випадку рівно ймовірних припущень.

Прийнявши до уваги ту обставину, що ІУФР є функціоналом від характеристик ОР, а також, розглядаючи лише репрезентативні обсяги навчальних вибірок, скористаємось наступними оцінками функціональної результативності ЕС у складі АСР:

$$AU_{1,m}^{(ls)}(cr)=\frac{C_{1,m}^{(ls)}}{n_{\min}};\quad AU_{2,m}^{(ls)}(cr)=\frac{C_{4,m}^{(ls)}}{n_{\min}},$$
$$mis1_{m}^{(ls)}(cr)=\frac{C_{2,m}^{(ls)}}{n_{\min}};\quad mis2_{m}^{(ls)}(cr)=\frac{C_{3,m}^{(ls)}}{n_{\min}};\qquad (2.23)$$

де $C_{1,m}^{(ls)}$ – кількість епізодів, коли встановлено, що реалізація ОР належить до КОН $C_{1,m}^{o}$, якщо дійсно $\{ct_1^{(j)}\}\in CT_1^o$;

$C_{2,m}^{(ls)}$ – кількість епізодів, коли встановлено, що реалізація ОР не належить до КОН $C_{1,m}^{o}$ якщо дійсно $\{ct_1^{(j)}\}\in CT_1^o$;

$C_{3,m}^{(ls)}$ – кількість епізодів, коли встановлено, що реалізації ОР належать до КОН $C_{1,m}^{o}$, якщо вони насправді належать класу CT_2^o;

$C_{4,m}^{(ls)}$ – кількість епізодів, коли встановлено, що реалізації ОР не належать до КОН $C_{1,m}^{o}$, якщо вони насправді належать класу CT_2^o;

$n_{\min}$ – мінімально потрібний обсяг репрезентативної вибірки для реалізації процедури навчання системи.

Таким чином, на підставі прийнятих оціночних параметрів функціональної результативності системи розпізнавання, та виконав підстановку характеристик (2.23) у вираз (2.19) отримаємо наступну формулу для обчислення ІУФР навчання АСР та ЕС ($CE_{1,m}^{(ls)}$), зокрема, розпізнаванню реалізацій класу кібератаки $-CT_1^o$:

$$CE_{1,m}^{(ls)} = 0{,}5\left(\frac{C_{1,m}^{(ls)}}{C_{1,m}^{(ls)} + C_{3,m}^{(ls)}} \cdot \log_2 \frac{C_{1,m}^{(ls)}}{C_{1,m}^{(ls)} + C_{3,m}^{(ls)}} + \right.$$
$$+ \frac{C_{2,m}^{(ls)}}{C_{2,m}^{(ls)} + C_{4,m}^{(ls)}} \cdot \log_2 \frac{C_{2,m}^{(ls)}}{C_{2,m}^{(ls)} + C_{4,m}^{(ls)}} + \frac{C_{3,m}^{(ls)}}{C_{1,m}^{(ls)} + C_{3,m}^{(ls)}} \cdot \log_2 \frac{C_{3,m}^{(ls)}}{C_{1,m}^{(ls)} + C_{3,m}^{(ls)}} + \quad (2.24)$$
$$\left. + \frac{C_{4,m}^{(ls)}}{C_{2,m}^{(ls)} + C_{4,m}^{(ls)}} \cdot \log_2 \frac{C_{4,m}^{(ls)}}{C_{2,m}^{(ls)} + C_{4,m}^{(ls)}} \right) + 1.$$

З метою отримання прийнятного для подальшого алгоритмічного використовування у ЕС рівняння для модифікованого критерію Кульбака – Лейблера, підставимо вираз (2.20) у вираз (2.18), та одержимо наступний результат:

$$CE = \frac{1}{n} \cdot \log_2 \left\{ 2n + 10^{-r} - \left[C_2^{(ls)} + C_3^{(ls)} \right] \Big/ \left[C_2^{(ls)} + C_3^{(ls)} \right] + 10^{-r} \right\} \cdot$$
$$\cdot \left\lfloor n - \left(C_2^{(ls)} + C_3^{(ls)} \right) \right\rfloor, \quad (2.25)$$

де r – кількість знаків у мантисі для $CE_m^{(ls)}$.

АСР що розробляється ґрунтується на обчисленні предикатів. У подібних системах, знання представляються за допомогою перекладу тверджень про ОР в формули логіки предикатів і додавання їх як аксіом в систему. Наприклад, розглянемо схему обчислення коефіцієнтів $C_2^{(ls)}$ і $C_3^{(ls)}$ у формулі (2.25). Предикатний вигляд обчислення коефіцієнтів $CT_2^{(ls)}$ і $CT_3^{(ls)}$ має наступний вигляд:

$$\begin{gathered}
\left(\forall\, CT_1{}^{o} \in RC^{|O|}\right)\left(\forall\, CT_2{}^{o} \in RC^{|O|}\right) \\
\left[if\ \ ct_1^{(j)} \in CT_1 \quad then\right. \\
C_1(j) := C_1(j-1) + 1 \\
\left.else\ \ C_2(j-1) + 1\right]; \\
\left(\forall\, CT_1{}^{o} \in RC^{|O|}\right)\left(\forall\, CT_2{}^{o} \in RC^{|O|}\right) \\
\left[if\ \ ct_2^{(j)} \in CT_1 \quad then\right. \\
C_3(j) := C_3(j-1) + 1 \\
\left.else\ \ C_4(j) := C_4(j-1) + 1\right]
\end{gathered}$$

Як показали результати аналізу попередніх досліджень по розробці СІРКЗ, багато авторів не приймають до уваги можливі так звані помилки під час завдання на прийняття рішення в ході процедур машинного навчання (МНав). Наприклад, коли приймається рішення «нелегітимне з'єднання», але не у результаті порівняння із «зразком», а через неможливість отримати адекватне представлення даних [70, 54, 80, 167, 193, 200, 210].

В роботах [26, 50–52, 137, 164, 188, 190] було запропоновано модифікувати ІУФР за мірою Кульбака – Лейблера виходячи з можливості застосування трьох альтернатив в системі оцінювання приймаємих рішень. Як зазначають автори, цей підхід має певні переваги перед традиційним

методом коли розглядаються лише дві альтернативи – «атака виявлена», або «атака невиявлена».

Зазначимо, що забезпечення стійкого функціонування на надійного опрацювання інформації у КВІС (КВКВ) в довільний момент часу в умовах впливу кібератаки атак досягається реалізацією відображення:

$$SO : SS \times CA \rightarrow SS_{res} = \{SS_{res}^{i}\}, \tag{2.26}$$

де SS_{res} – множина дозволених станів КВІС (КВКВ);

$CA = \{CA_0, CA_1, ..., CA_N\}$ – множина реалізації кібератак.

Така реалізація забезпечується організацією взаємопов'язаних процесів регулювання параметрів КВІС (КВКВ) CO, методів і моделей протидії кібератакам CM, комплексу засобів попередження, розпізнавання, аналізу кібератак, та активної протидії атакам ME. У цьому випадку функціонал, що визначає узагальнений показник ефективності протидії кібератакам, враховує показник ефективності розпізнавання, а також, характеризує стійкість функціонування КВІС (КВКВ) подамо так:

$$IE = F[(SCA, CE), (SS, T_s, VIL), (CO, CM, ME], \tag{2.27}$$

де SCA – сценарії кібератак;

CE – критерій ефективності розпізнавання ОР;

SS - множина параметрів КВІС (КВКВ):

T_s – періоди часу виконання функціональних завдань у КВІС (КВКВ);

VIL – уразливості ПЗ та ІМ КВІС (КВКВ);

множина параметрів протидії загрозам та кібератакам:

CO – параметри регулювання КВІС (КВКВ);

CM – методи протидії загрозам та кібератакам у КВІС (КВКВ);

ME – засоби попередження, виявлення, аналізу та активної протидії кібератакам.

З метою визначення яким чином інформаційна міра Кульбака – Лейблера залежить від параметрів ЕС для варіанту застосування керуючих команд, які ґрунтуються на трьох альтернативах (випадок коли приймається рішення про динаміку змінювання параметра IE), введемо такі гіпотези:

1) Основна робоча гіпотеза яка (базова) – hy_{γ_1} : реалізація ознаки rc_i ОР (*RS*) та показник IE знаходиться у межах звичайного стану КВІС;

2) гіпотеза hy_{γ_2} – реалізацій ознаки rc_i ОР (*RS*) та показник IE дозволяють зробити висновок, що значення показника IE є меншими на норму;

3) гіпотеза hy_{γ_3} – показник IE дозволяють зробити висновок, що значення показника IE є більшими за норму.

Відповідно до прийнятих припущень позначимо апостеріорні гіпотези так: hy_{μ_1} – значення реалізацій ознак належить полю допустимих відхилень (ПДВ) ca, hy_{μ_2} – значення реалізацій ознак знаходиться лівіше за ПДВ ca; hy_{μ_3} – значення реалізацій ознак знаходиться правіше за ПДВ ca.

Враховуючи попередні викладки, для рішення АСР яке допускає три альтернативи, отримаємо наступні характеристики, див. табл. 2.6.

Таблиця 2.6

Характеристики точності розпізнаванні у АСР для трьох прийнятих альтернатив

№	Параметр	Позначення	Опис
1	перша валі-дація гіпотези (ВАЛГ)	$AU_{1,m}^{(ls)} = p\left(hy_{\gamma_1} / hy_{\mu_1}\right)$	перша ВАЛГ базується на висновках
2	друга ВАЛГ	$AU_{2,m}^{(ls)} = p\left(hy_{\gamma_2} / hy_{\mu_2}\right)$	друга ВАЛГ базується на порівнянні відхилень від $\left\{ca_{K,i}^{*}\right\}$
3	третя ВАЛГ	$AU_{3,m}^{(ls)} = p\left(hy_{\gamma_3} / hy_{\mu_3}\right)$	третя ВАЛГ базується на результатах опрацювання коефіцієнтів $CT_2^{(ls)}$ і $CT_3^{(ls)}$
4	перша помилка першого роду	$mis1_{1,m}^{(ls)} = p\left(hy_{\gamma_2} / hy_{\mu_1}\right)$	Кількість помилкових спрацювань АСР або ЕС в процесі виявлення кібератак
5	друга помилка першого роду	$mis1_{2,m}^{(ls)} = p\left(hy_{\gamma_3} / hy_{\mu_1}\right)$	
6	перша помилка другого роду	$mis2_{1,m}^{(ls)} = p\left(hy_{\gamma_1} / hy_{\mu_2}\right)$	Кількість невиявлених в ході роботи АСР або ЕС з виявлення кібератак
7	друга помилка другого роду	$mis2_{2,m}^{(ls)} = p\left(hy_{\gamma_3} / hy_{\mu_2}\right)$	
8	перша помилка під час завдання на прийняття рішення в ході процедур МНав	$mis3_{1,m}^{(ls)} = p\left(hy_{\gamma_1} / hy_{\mu_3}\right)$	Помилки під час завдання на прийняття рішення в ході процедур МНав, коли модель не враховує деякі елементи МІЕТ [80]

Продовження таблиці 2.6

9	друга помилка під час завдання на прийняття рішення в ході процедур МНав	$mis3_{2,m}^{(ls)} = p\left(hy_{\gamma_2} / hy_{\mu_3}\right)$	

Будемо вважати, що: характеристики $mis2_{2,m}^{(ls)}$ та $mis3_{2,m}^{(ls)}$ є малоймовірними, тому їх можна не приймати до уваги. Також приймаємо:

$$\begin{aligned} mis1_{m}^{(ls)} &= mis1_{1,m}^{(ls)} = mis1_{2,m}^{(ls)}; \\ mis2_{m}^{(ls)} &= mis2_{1,m}^{(ls)}; \\ mis3_{m}^{(ls)} &= mis3_{1,m}^{(ls)}. \end{aligned} \tag{2.28}$$

Обчислюємо повні ймовірності $P_{t,m}^{(ls)}$ і $P_{f,m}^{(ls)}$ з урахуванням припущень (2.28)

$$P_{t,m}^{(ls)} = p\left(hy_{\mu_1}\right)AU_{1,m}^{(ls)} + p\left(hy_{\mu_2}\right)AU_{2,m}^{(ls)} + p\left(hy_{\mu_3}\right)AU_{3,m}^{(ls)}; \tag{2.29}$$

$$P_{f,m}^{(ls)} = p\left(hy_{\mu_1}\right)mis1_{m}^{(ls)} + p\left(hy_{\mu_2}\right)mis2_{m}^{(ls)} + p\left(hy_{\mu_3}\right)mis3_{m}^{(ls)}.$$

Згідно з принципом Бернуллі-Лапласа для трьох прийнятих гіпотез можна вважати, що, $p\left(hy_{\mu_1}\right) = p\left(hy_{\mu_2}\right) = p\left(hy_{\mu_3}\right) = 1/3$.

Тоді підставив вирази (2.28) і (2.29) в формулу (2.21) отримаємо наступний результат

$$AU_{1,m}^{(ls)} + 2mis1_m^{(ls)} = 1;\ AU_{2,m}^{(ls)} + mis2_m^{(ls)} + mis3_m^{(ls)};$$
$$AU3_{3,m}^{(ls)} + mis2_m^{(ls)} + mis3_m^{(ls)}. \qquad (2.30)$$

Враховуючи вираз (2.31), представимо формулу (2.30) для обчислення $CE_m^{(k)}$ АСР у вигляді

$$CE_m^{(ls)} = 1/3 \cdot \left\{AU_{1,m}^{(ls)} + 1 - 2\left[mis2_m^{(ls)} + mis3_m^{(ls)}\right]\right\} \cdot$$
$$\cdot \log_2 \frac{2AU_{1,m}^{(ls)} + 4 - 4 \cdot \left[mis2_m^{(ls)} + mis3_m^{(ls)}\right]}{1 - AU_{1,m}^{(ls)} + 2 \cdot \left[mis2_m^{(ls)} + mis3_m^{(ls)}\right]} \quad . \qquad (2.31)$$

Таким чином, отриманий вираз (2.31) який враховує модифіковані ентропійний критерій та міру Кульбака – Лейблера є функціоналом від характеристик рішень, які приймаються під час розпізнавання відповідних кібератак на КВІС. Окрім того, вираз (2.31) враховує відомі статистичні й детерміновані (дистанційні) критерії оптимізації процедури кластеризації реалізацій ознак ОР на попередньому етапі функціонування здатних навчатися систем виявлення кібератак.

Вирішальне правило визначає віднесення вектору параметрів реалізації відомих або невідомих сценаріїв кібератак SCA_m^{CT} m-го об'єкту та ct-го класу до одного з відомих класів ОР $RS_{m_j}^{CT}$ на j-му кроці роботи засобів кіберзахисту. Відповідно до критерію Байеса вирішальне правило виглядає наступним чином:

$$P\left(RS_{m_i}^{CT}\right)\cdot P\left(\overline{SCA_m^{CT}}/RS_{m_i}^{CT}\right)\geq P\left(RS_{m_k}^{CT}\right)\cdot P\left(\overline{SCA_m^{CT}}/RS_{m_k}^{CT}\right), \quad (2.32)$$

де $P\left(RS_{m_i}^{CT}\right)$ – ймовірність віднесення АСР ОР (кібератаки) до класу відомих ОР $RS_{m_i}^{CT}$;

$P\left(\overline{SCA_m^{CT}}/RS_{m_i}^{CT}\right)$ – щільність умовної ймовірності віднесення АСР виявленого ОР до відомого класу $RS_{m_i}^{CT}$;

$P\left(RS_{m_k}^{CT}\right)$ – ймовірність віднесення АСР ОР до класу невідомих ОР $RS_{m_k}^{CT}$;

$P\left(\overline{SCA_m^{CT}}/RS_{m_k}^{CT}\right)$ – щільність умовної ймовірності віднесення АСР виявленого ОР до невідомого класу $RS_{m_k}^{CT}$.

Також, на основі критерію Байеса визначаємо середню «ціну» ризику прийняття у АСР рішення щодо віднесення вектору параметрів невідомих ОР до класу $RS_{m_k}^{CT}$:

$$PR\left(RUL_i/\overline{SCA_m^{CT}}\right)=\sum_{j=1}^{\gamma} np\left(\frac{RUL_i}{RS_{m_k}^{CT}}\right)\cdot P\frac{RS_{m_k}^{CT}}{\overline{SCA_m^{CT}}}, \quad (2.33)$$

де RUL_i – вирішальне правило, за яким бінарний навчальний вектор (БНВ) ОР $\overline{SCA_m^{CT}}$ визначає приналежність об’єкту до $RS_{m_k}^{CT}$;

$np\left(\frac{RUL_i}{RS_{m_k}^{CT}}\right)$ – умовна «ціна» прийняття АСР рішення RUL_i;

$P\dfrac{RS_{m_k}^{CT}}{\overline{SCA_m^{CT}}}$ – умовна ймовірність того, що $\overline{SCA_m^{CT}}$ віднесений АСР до класу $RS_{m_k}^{CT}$.

Тестування ЕС із визначенням належності реалізації класу CT_m^o до ОР здійснюється за наступним вирішальним правилом:

$$if\left(1-\frac{\sum_{1}^{N}\left(\mathrm{cov}_{m,i}\oplus\mathrm{cov}_{e,i}\right)}{cr_{1,m}^{*}}\geq 0\right)then$$

$$hy_m=1-\frac{\sum_{i=1}^{N}\left[\left(\mathrm{cov}_{m,i}\oplus\mathrm{cov}_{e,i}\right)\cdot cop_{m,i}^{*}\right]}{cr_{2,m}^{*}},$$

де $\mathrm{cov}_{e,i}$ – значення $i-й$ координати отриманої під час тестування;

$cr_{1,m}^{*}$ – оптимальний радіус КОН класу кібератаки для CT_m^o;

$cop_{m,i}^{*}-i-me$ значення координати вектора, який є ортогональним вісі, що проходить через вершину еталонного CT_m;

$cr_{2,m}^{*}$ – оптимальний радіус КОН класу кібератаки для CT_m^o.

Для випадку коли АСР виконує порівняльний аналіз двох БНМ, як було попередньо показано на рис. 2.3, вирішальне правило із використанням критерію Байеса можна записати у вигляді наступного співвідношення:

$$\frac{P\left(\overline{SCA_m^{CT}}\Big/RS_{m_1}^{CT}\right)}{P\left(\overline{SCA_m^{CT}}\Big/RS_{m_2}^{CT}\right)}\geq\frac{P\left(RS_{m_2}^{CT}\right)}{P\left(RS_{m_1}^{CT}\right)}. \qquad (2.34)$$

Проілюструємо яким чином з метою побудови правил RUL в процесі навчання АСР, формується апріорно нечітка класифікована навчальна матриця. Припустимо, що відома апріорно некласифікована багатовимірна навчальна матриця для СІРКЗ $\left\| lm_i^{(j)} \right\|, \; i = \overline{1, N}, \; j = \overline{1, n}.$ Сформулюємо задачу наступним чином:

1) в режимі кластерного аналізу необхідно перетворити вхідну навчальну матрицю реалізацій ознак у нечітку класифіковану;

2) в режимі навчання побудувати чітке розбиття простору реалізацій ознак ОР на класи $\left\{ CT_c^0 \mid m = \overline{1, M} \right\}$, які відповідно характеризують функціональні стани керованого процесу кіберзахисту, шляхом оптимізації координат вектора параметрів функціонування системи ІБ для КВІС

$$is = \left\langle M, fuz, ca, ct_{m1}, ct_{m2}, RS_b, cr_m \right\rangle, \tag{2.35}$$

де M – кількість кластерів або потужність алфавіту класів розпізнавання кібератак;

fuz – показник нечіткості для алгоритму;

ct_{m1}, ct_{m2} – двійкові вектори, що визначають координати першого та другого фокусів КОН для класу кібератак в бінарному просторі реалізацій ознак кібератак RS_b ;

cr_m – піввісь к КОН класу в просторі реалізацій ознак кібератак RS_b.

Введемо такі обмеження

$$\begin{cases} 2 \le M \le n / n_{\min}; \\ n_c \ge n_{\min}; \\ fuz > 1; \\ cr_m > c_m,\ c_m \le N / 2; \\ cr(ct_{c1} \oplus ct) + cr(ct_{c2} \oplus ct) - cr_d > 0, \\ \forall ct \in \{ct : cr(ct_{m1} \oplus ct) + cr(ct_{m2} \oplus ct) = 2cr_m\}; \\ ca \in [0,\ ca_n / 2], \end{cases} \tag{2.36}$$

де $n_{\min}$ – мінімальний обсяг навчальної вибірки для кожного класу реалізацій ознак кібератак (вибірка обов'язково повинна бути репрезентативною);

n_c – кількість реалізацій в межах класу CT_c^0;

$cr(ct_{c1} \oplus ct)$, $cr(ct_{c2} \oplus ct)$ – відповідно, кодові відстані від першого та другого фокусів КОН сусіднього класу CT_c^0;

cr_d – відстань до центра КОН в межах класу CT_c^0 в просторі реалізацій ознак RS_b;

$cr(ct_{m1} \oplus ct)$, $ct(ct_{m2} \oplus ct)$ – відповідно, кодові відстані від першого та другого фокусів КОН класу CT_c^0;

ca_n – нормоване поле допустимих відхилень.

В процесі навчання СІРКЗ визначаються координати вектора параметрів терму (2.33) при обмеженнях (2.34). Це, у свою чергу, дає змогу забезпечити max значення усередненої за алфавітом класів ІУФР розпізнавання кібератак на КВІС, скориставшись виразом (2.32), відповідно:

$$\overline{CE}^* = (1 \Big/ M) \cdot \sum_{m=1}^{M} \max_{\{ls\}} CE_c ,$$

де CE_c – значення ІУФР навчання СІРКЗ для реалізації класу кібератак – CT_c^0;

$\{ls\}$ – множина кроків для навчанні СІРКЗ.

У режимі тестової перевірки СІРКЗ приймається рішення про належність реалізацій еталонних образів, що характеризують поточний функціональний стан інформаційної безпеки, до відповідного класу із сформованого на етапі навчання СІРКЗ алфавіту. Тобто, на цьому етапі виконується дефазифікація даних – $\left\{CT_c^0 \mid c = \overline{1, C}\right\}$.

Ініціалізація вхідних некластеризованих даних про реалізацій ознаки кібератак подано у вигляді (векторної) матриці $\left\{lm_i^{(j)}, \mid i = \overline{1, N},\ j = \overline{1, n}\right\}$.

На наступному етапі роботи алгоритму генеруються матриці нечіткого розбиття:

$$V = \begin{bmatrix} v_{11}\ v_{12} \ldots v_{1n} \\ v_{21}\ v_{22} \ldots v_{2n} \\ \ldots \\ v_{M1}\ v_{M2} \ldots v_{Mn} \end{bmatrix} \quad \text{за умов}$$

$$v_{mj} \in \{0,1\}; \sum_{m=1}^{M} v_{mj} = 1; 0 < \sum_{j=1}^{n} v_{mj} < n, \tag{2.37}$$

де v_{mj} – ступінь належності *j*-го об’єкта до кластеру – *m*.

Розрахунок центрів кластерів реалізацій ознак кібератак здійснюється за наступною формулою:

$$lm_m = \sum_{j=1}^{n} (v_{mj}^{(l-1)})^{fuz} \cdot lm^{(j)} \Big/ \sum_{j=1}^{n} (v_{mj}^{(l-1)})^{fuz}, \tag{2.38}$$

де l – лічильник кількості ітерацій в процесі навчання АСР.

У результаті роботи алгоритму мінімізується цільова функція:

$$OFU = \sum_{m=1}^{M} \sum_{j=1}^{n} v_{mj}^{fuz} \cdot cr_{K^{(m)}}^{2}(lm^{(j)}, lm_c), \tag{2.39}$$

де $K^{(m)} = \dfrac{\sum_{j=1}^{n} (v_{mj}^{(l-1)})^{fuz} \cdot (lm^{(j)} - lm_m)^T \cdot (lm^{(j)} - lm_c) \cdot (lm_m - lm_c)}{\sum_{j=1}^{n} (v_{mj}^{(l-1)})^{fuz}}$

– коваріація для кластеру – m;

T – множина моментів часу зняття інформації.

У разі потреби, виконується переобчислення елементів матриці нечіткого розбиття виконується за наступною формулою:

$$v_{mj}^{(l)} = 1 \Big/ \left[\sum_{k=1}^{M} \left(\frac{cr_{K^{(c)}}^{2}(lm^{(j)}, lm_c)}{cr_{K^{(c)}}^{2}(lm^{(j)}, lm_w)} \right)^{\frac{1}{fuz-1}} \right]. \tag{2.40}$$

Перевірку моделі виконано для 5 класів поширених кібератак на КВІС – «відмова в обслуговуванні», «завантаження ворожого ПЗ», «несанкціоноване виконання команд», «порушення прав доступу», «несанкціонований доступ до паролю».

При цьому кількість реалізацій ознак розпізнавання варіювалась в межах $N = 9 - 15$. Оптимальна кількість кластерів обиралась за max ІУФР навчання СІРКЗ. Як показав аналіз результатів, оптимальна кількість кластерів дорівнює 3.

Наприклад, розглянемо зміну ентропії системи у результаті мережевої кібератаки класу «відмова в обслуговуванні» (DoS атака). Ентропія КВІС визначається співвідношенням:

$$H_{nv} = -\sum_{ib=\min}^{\max} P_{ib} \cdot \log_2 P_{ib},$$

де nv – можливе число варіантів довжин пакета під час атаки;

P_{ib} – імовірність появи в каналі передачі даних пакета довжиною *ib* байт;

min – мінімальне значення довжини пакета;

max – максимальне значення довжини пакета.

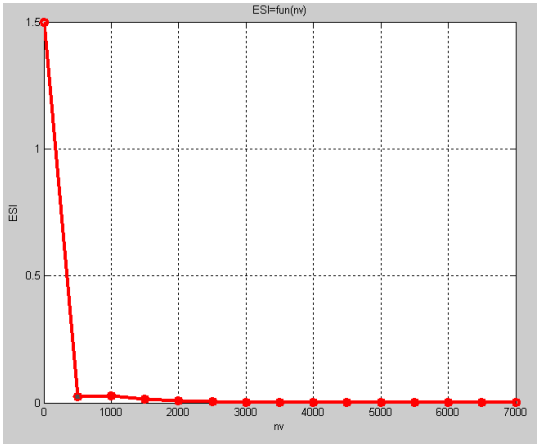
При відсутності вторгнення ентропія системи при більших nv приблизно постійна величина, що рівна 4 [188, 195, 197].

За допомогою генератора трафіка для імітаційної моделі сегменту ЛОМ КВІС у пакеті MATLAB(Simulink) створювалися атаки 1-ої і 2-ої групи. У першій групі моделювалися атаки, при яких використовуються мережні пакети однакової довжини. У другій групі розглядалися атаки, при яких довжина пакета - стохастична величина в заданому інтервалі.

На рис. 2.7 наведена крива зміни ентропії залежно від довжини переданих мережних пакетів. Із графіка бачимо, що при збільшенні числа nv ентропія системи при впливі атакою 1-ої групи швидко прагне до 0. Це виділяє DoS – атаку 1-ої групи із процесів, що відбуваються в мережі.

На рис. 2.8 зображена ентропія для DoS – атаки 2-ої групи. Ентропія також прагне до 0, але повільніше, ніж у першому випадку. Це пов'язано з тим, що ця DoS – атака здійснювалася потоком мережних пакетів випадкової довжини, рівномірно розподіленої на деякому інтервалі.

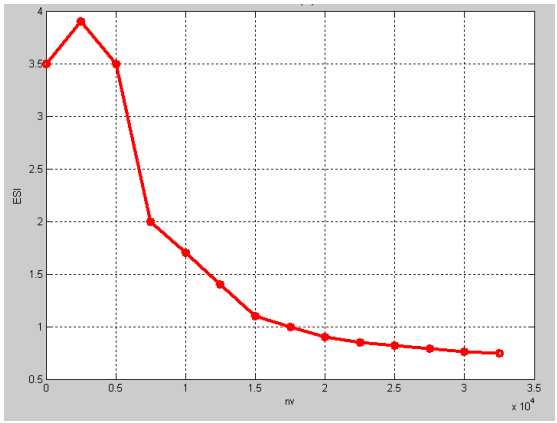
H_{nv}



nv

Рис. 2.7. Зміна ентропії системи залежно від довжини переданих мережних пакетів $H = f(nv)$ для кібератак першої групи

H_{nv}



nv

Рис. 2.8. Зміна ентропії системи залежно від довжини переданих мережних пакетів $H = f(nv)$ для кібератак другої групи

У цілому результати моделювання свідчать про працездатність розглянутого підходу й можливості розпізнавання станів КВІС за прийнятний час і, практично, з гарантованою надійністю, під якою розуміємо нижній поріг числа помилок розпізнавання.

Отже, можна припустити, що порівнянням кривих для випадків “типового” (відсутність атаки) і змішаного (наявність атаки на тлі “типового”) трафіків можна виявити наявність атаки. Таким чином, порівнювані залежності можуть бути додатковою ознакою для АСР.

Оскільки кожний стан системи може характеризуватися сукупністю значень квантованих цифрових сигналів, характерних для SS_i, то, у термінах

АРС, число градацій реалізацій ознаки – рівня квантування, виступає як універсальна ББНМ, потужність якої дорівнює максимальному рівню квантування, характерному для даної моделі.

2.3. Висновки до другого розділу

Результатом проведених у даному розділі досліджень стали такі висновки:

1) розроблено модель адаптивної експертної системи у складі інтелектуальних систем виявлення вторгнень, у якій, на відміну від існуючих, застосовується процедура нечіткої кластеризації реалізацій ознак кібератак та корекції вирішальних правил, що дозволяє створювати адаптивні механізми самонавчання систем інтелектуального розпізнавання кібератак;

2) запропоновано для оцінки якості розбиття простору реалізацій ознак ОР у АСР застосовувати в якості оціночного показника модифіковану інформаційну умову функціональної ефективності (ІУФР), здатної до самонавчання системи розпізнавання;

3) доведено, що застосування моделей та алгоритмів кластерізації реалізацій ознак ОР, які ґрунтуються на ентропійному та інформаційно-дистанційному критерії Кульбака – Лейблера, дозволяє отримувати вхідну нечітку класифіковану навчальну матрицю яка використовується як об’єкт навчання, та в рамках інтелектуальних технологій та алгоритмів навчання АСР будувати коректні вирішальні правила розпізнавання кібератак на КВІС;

4) встановлено, що збільшення кількості векторів–реалізацій класів ОР при виявлення кібератак на КВІС призводить до збільшення значення максимального значення інформаційної умови функціональної результативності, а також дозволяє отримувати коректні правила для адаптивної здатної до самонавчання системи розпізнавання;

5) доведено, що оцінка якості розбиття простору реалізацій ознак кібератак та інших варіантів легітимного втручання в роботу КВІС, може

бути ефективно вирішена на основі ІУФР та процедури корекції вирішальних правил розпізнавання, що дозволяє зменшити кількість попередньої інформації яка підлягає опрацюванню аналітиками служб інформаційної безпеки КВІС;

6) показано, що оптимальна кількість кластерів для визначення max значення ІУФР при навчанні АСР, дорівнює 3.

РОЗДІЛ 3
МЕТОДИ КЛАСТЕРИЗАЦІЇ ТА ОПТИМІЗАЦІЇ КОНТРОЛЬНИХ ДОПУСКІВ НА РЕАЛІЗАЦІЇ ОЗНАКИ РОЗПІЗНАВАННЯ КІБЕРАТАК

3.1. Метод кластеризації реалізацій ознак при виявлення складних кібератак

Процедуру розбиття простору реалізацій ознак (ПОЗ) та наступної кластеризації, для будь-якого класу ОР CT_m^o, у відповідності до [50-52, 80, 94, 137, 148, 151, 164, 169, 188, 190, 209], пропонується здійснювати шляхом перетворення ПОЗ у прийнятну для парадигми ООП форму. Оскільки основним етапом кластеризації під час розбиття ПОЗ на групи є збільшення радіусу (cr_m) контейнера (РКОН) на кожному кроці навчання, можна скористатися наступним рекурентним виразом:

$$cr_m(ls) = \left[cr_m(ls-1) + \xi \mid cr_m(ls) \in IS_m^{cr} \right], \qquad (3.1)$$

де ls – кількість кроків збільшення радіусу КОН C_m^o;

ξ – прийняті для вибраних реалізацій ознак кроків збільшення КОН;

IS_m^{cr} –допустима величина радіуса КОН.

Не виділяючи окремих деталей, при навчанні ЕС приймається припущення про нечітку компактність реалізації БНМ, отриманих на етапі розбиття ПОЗ на відповідні класи ОР. Також, припускаємо, що нечітке розбиття $RC^{|M|}$ включає елементи які можна віднести до нечітких класів ОР.

Для побудови в ході навчання ЕС коректних правил, згідно із принципами, викладеними у роботах Івахненка О. Г. [61], Карпинского Н.П.

[62], Calleri [137], Kabiri P. [162], а також [26, 50, 80, 94 та ін.], за багатоциклічною процедурою пошуку максимальної граничної величини інформаційного показника (умови) функціональної результативності (ІУФР) навчання ЕС, подамо так:

$$is'_k = \arg\max_{IS_k}\{\max_{IS_{k-1}}\{\ldots\{\max_{IS_1 \cap IS_{CE}} \frac{1}{M}\sum_{m-1}^{M} CE_m\}\ldots\}\}, \tag{3.2}$$

де CE_m – ІУФР навчання ЕС розпізнавати ОР які належать класу CT_m^o ;

IS_k – допустимий діапазон значень $k-ї$ інформативної ознаки ОР;

IS_{CE} – допустимий діапазон величини ІУФР в ході навчання ЕС.

На вираз (3.2) в ході «тренування» ЕС у складі СІРКЗ накладаються наступні обмеження:

$$\left(\forall CT_m^o \in RC^{|M|}\right)\left\lfloor CT_m^o \neq \varnothing \right\rfloor; \tag{3.3}$$

$$\left(\exists CT_a^o \in RC^{|M|}\right)\left(\exists CT_b^o \in RC^{|M|}\right)\left\lfloor \begin{matrix} CT_a^o \neq CT_b^o \rightarrow \\ \rightarrow CT_a^o \bigcap CT_b^o \neq \varnothing \end{matrix} \right\rfloor; \tag{3.4}$$

$$\left(\forall CT_a^o \in RC^{|M|}\right)\left(\forall CT_b^o \in RC^{|M|}\right)\left\lfloor \begin{matrix} CT_a^o \neq CT_b^o \rightarrow \\ \rightarrow B\,CT_a^o \bigcap B\,CT_b^o = \varnothing \end{matrix} \right\rfloor, \tag{3.5}$$

де $B\,CT_a^o$, $B\,CT_b^o$ ядра класів ОР CT_a^o і CT_b^o, відповідно;

$$\bigcup_{CT_m^o \in RC} CT_m^o \subseteq RS_B; a \neq b; a, b, m = \overline{1, M} \tag{3.6}$$

Зробимо припущення, що класи CT_a^o та CT_b^o є сусідніми та мають мінімальну відстань між центрами кластерів $cr(ct_a \oplus ct_b)$ серед усіх класів для ОР, які описані відповідними БНМ, де ct_a і ct_b – еталонні вектори класів ОР.

Таким чином, за відповідною інтелектуальною технологією навчання ЕС (АСР), та попереджаючи випадки поглинання одним класом базових реалізацій ознак серцевини іншим класом, що визначається умовами (3.3) – (3.6), процедуру навчання можна подати у вигляді наступного предикатного виразу:

$$\left(\forall CE_a^o \in RC^{|M|}\right)\left(\forall CT_b^o \in RC^{|M|}\right)\left[\begin{array}{l} CT_a^o \neq CT_b^o \rightarrow \\ \rightarrow \left(cr_a' < cr\left(ct_a \oplus ct_b\right)\right)\cdot \\ \cdot \left(cr_b' < cr\left(ct_a \oplus ct_b\right)\right) \end{array}\right], \qquad (3.7)$$

де cr_a' , cr_b' – оптимальні радіуси КОН C_a^o і C_b^o, відповідно.

Для зменшення кількості циклів під час процедури навчання, потрібно визначити множини вхідних сигналів (факторів), які діють на ЕС, та корелюються з розмірністю вектора параметрів «тренування» $is =< is_1, \ldots, is_k, \ldots, is_{RS} >$ під час розпізнавання та кібератак атак «шаблонів».

Під час навчання ЕС в модулях МНЕС та МТЕС (рис. 2.6 та табл. 2.5) приймається рішення про належність реалізацій ОР до одного класів $\{CT_m^o\}$.

Метод навчання ЕС у складі СІРКЗ являє собою ітераційну процедуру пошуку глобального ІУФР в допустимому діапазоні визначення його функції [50-52, 80, 94, 137, 148, 151, 164, 169, 188, 190, 209]:

$$ca^* = \arg \max_{IS_{ca}} \{ \max_{IS_{CE} \cap IS_{cr}} \overline{CE} \} , \qquad (3.8)$$

де IS_{ca} – допустимий діапазон величин контрольних відхилень ca для класу ОР $\{CT_m^o\}$;

IS_{CE} – робочий діапазон визначення показника ІУФР $\overline{CE}$;

IS_{cr} – допустимий діапазон величини РКОН cr.

Реалізація методу (алгоритму класифікації) здійснюється при наступних обмеженнях

$$\left(\forall CT_{m,\xi}^{o} \in RC^{|M|}\right)\left\lfloor CT_{m,\xi}^{o} \neq \varnothing,\ m = \overline{1, M}\right\rfloor, \tag{3.9}$$

$$\left(\forall CT_{m,\xi}^{o} \in RC^{|M|}\right)\left(\forall CT_{c,\xi}^{o} \in RC^{|M|}\right)\left\lfloor \begin{array}{l} CT_{m,\xi}^{o} \neq \\ \neq CT_{c,\xi}^{o} \rightarrow BCT_{m,\xi}^{o} \cap BCT_{c,\xi}^{o} = \varnothing \end{array}\right\rfloor, \tag{3.10}$$

$$\left(\forall CT_{m,\xi}^{o} \in RC^{|M|}\right)\left(\forall CT_{c,\xi}^{o} \in RC^{|\mathrm{O}|}\right)\left[\begin{array}{l} CT_{m,\xi}^{o} \\ \neq CT_{c/\xi}^{o} \rightarrow \\ \left(cr'_{m,\xi} < cr\left(ct_{m,\xi} \oplus ct_{c,\xi}\right)\right) \wedge \\ \wedge \left(cr'_{c,\xi} < cr\left(ct_{m,\xi} \oplus ct_{c,h}\right)\right) \end{array}\right], \tag{3.11}$$

$$\bigcup_{\mathrm{CT}_{\mathrm{m},\xi}^{\mathrm{o}} \in RC} CT_{m,\xi}^{o} \subseteq RS, \tag{3.12}$$

де $B\,CT_{m,\xi}^{o}$, $B\,CT_{c,\xi}^{o}$ – центри двох найближчих кластерів $CT_{m,\xi}^{o}$ та $CT_{c,\xi}^{o}$, відповідно;

ξ – крок приріста значення радіуса КОН;

$cr'_{m,\xi}$, $cr'_{c,\xi}$ – відповідно, оптимізовані радіуси КОН $CT^{o}_{m,\xi}$ та $CT^{o}_{c,\xi}$;

$cr\left(ct_m \oplus ct_c\right)$ – міжцентрова кодова відстань кластерів $CT^{o}_{m,\xi}$ та $C^{o}_{c,\xi}$.

Розглянемо основні етапи реалізації методу розбиття ПОЗ на кластери в ході реалізації процедури розпізнавання ОР. Для наочності етапи розбиття простору реалізацій ознак ОР на кластери подамо у табличному вигляді, див. таблиця 3.1.

Таблиця 3.1

Етапи розбиття ПОЗ на кластери

№	Етап	Дія	Математичний опис для методу кластеризації реалізацій ознак ОР
1	1	Значення лічильника кроків зміни перевірочних допустимих відхилень (ПДВ) ca_i на реалізацій ознаки ОР виставляється рівним "0":	$l := 0$.
2	2	Обчислення нижнього $A_{low_i}[l]$ та верхнього $A_{up_i}[l]$ ПДВ реалізацій ознак ОР для всього простору RS:	$A_{low_i}[l] = lm_i - ca\dfrac{ca_{low_i}}{100}$; $A_{up_i}[l] = lm_i + ca\dfrac{ca_{low_i}}{100}$, де lm_i – i-та ознака еталонного вектора-реалізації некласифікованої багатовимірної навчальної матриці (НБНМ) $\left\|lm_i^{(j)}\right\|$; ca_{low_i} – перевірочні допустимі відхилення для ознак ОР, які визначаються на основі методів мультифрактального аналізу, показника Херста, рухомого вікна [132, 190, 200 та ін.].

продовження таблиці 3.1

3	3	Значення лічильника кроків зміни перевірочних допустимих відхилень (ПДВ) ca_i на реалізацій ознаки ОР збільшується на 1:	$l := l + 1$
4	4	Формування бінарної навчальної матриці (БНМ) $\left\|ct_i^{(j)}\right\|$:	Правило $ct_i^{(j)} = \begin{cases} 1, if\ A_{low_i}[l] < lm_i^{(j)} < A_{up_i}[l]; \\ 0, else. \end{cases}$
5	5	Ініціалізація показника значення лічильника кроків для збільшення радіусу КОН:	$\xi := 1$.
6	6	Виконується процедура розбиття вхідної навчальної матриці ОВН $\left\{ct_i^{(j)}\right\}$ на два кластери:	$\left\{CT_m^o[\xi] \mid m = \overline{1,2}\right\}$.
	6.1	Обчислюються початкові еталонні вектори для реалізацій ознак ОР $\{ct_m\}$ для кластерів CT_m^o :	Перевірка умов: 1) $cr(ct_1 \oplus ct^0) \to \min$, $cr(ct_2 \oplus ct^1) \to \min$; 2) $cr(ct_1 \oplus ct_2) \to \max$, де ct^0 , ct^1 – нульовий та одиничний вектори.
	6.2	Значення радіусів кластерів CT_m^o виставляється рівним "0" :	$cr_m[\xi] := 0, n_m := 0$, де n_m – кількість реалізацій ОР, які належать кластеру CT_m^o .
	6.3	Визначаються реалізації для ОР, які належать кластерам $CT_m^o[\xi]$:	Правило: $ct_i \in CT_1^o[\xi],\ if\ cr(ct_i \oplus ct_1) <= <= cr\ \&\ cr(ct_i \oplus ct_1) < (ct_i \oplus ct_2);$ $ct_i \in CT_2^o[\xi],\ if\ cr(ct_i \oplus ct_2) <= <= cr\ \&\ cr(ct_i \oplus ct_2) < (ct_i \oplus ct_1);$ де $ct_i \mid i = \overline{1,N}$ – реалізації БНМ $\left\|ct_i^{(j)}\right\|$.

продовження таблиці 3.1

	6.4	Обчислення значення ІУФР:	$\overline{CE}^{*} = (1/M) \cdot \sum_{m=1}^{M} \max_{\{ls\}} CE_c$, де CE_c – значення ІУФР навчання АЕС для реалізації класу кібератак – CT_c^0; $\{ls\}$ – множина кроків для навчанні АЕС у складі СІРКЗ. М – кількість керуючих команд
	6.5	Формування множини $\{ct_m\}$ еталонних реалізацій для кластерів $\{CT_m^o[\xi]\}$:	Правило для визначення координат: $ct_{m,i} = \begin{cases} 1, \ if \ \frac{1}{n}\sum_{j=1}^{n} cr_{m,i}^{(j)} > 1/2; \\ 0, else. \end{cases}$
	6.6	Збільшення радіусу (РКОН):	$cr_m[\xi] := cr_m[\xi] + 1$
	6.7	Обчислення значення ІУФР та оптимальних радіусів кластерів $\{CT_m^o[\xi]\}$:	За умов: $N' = \sum_{m=1}^{M} n_m < N$, де N' – кількість реалізацій ОР яка належать RC_ξ та $cr_m[\xi] < cr(ct_1 \oplus ct_2)$.
7	7	Збільшується значення лічильника кроків:	$\xi := \xi + 1.$
8	8	Виконується процедура розбиття бінарного простору ознак (БПОЗ) на 3 кластери:	$\{CT_m^o[\xi] \mid m = \overline{1,3}\}.$
	8.1	Обчислюється бінарна матриця для кластера CT_3^o, еталонний вектор-реалізація ct_3 якого задовольняє умовам:	Перевірка умов: $cr(ct_1 \oplus ct_3) \to \min \ \& \ cr(ct_2 \oplus ct_3) \to \min$, де ct_1, ct_2 – еталонні реалізації кластерів $\{CT_m^o \mid m = \overline{1,2}\}$, відновлених при виконанні етапу 6.
	8.2	Значення радіусів кластеру CT_3^o виставляється рівним "0":	$cr_3[\xi] := 0.$

продовження таблиці 3.1

	8.3	Визначення випадків потрапляння реалізацій ознак ОР у кластер CT_3^o:	Правило для визначення випадків потрапляння реалізацій ознак ОР у кластер CT_3^o: $ct_i \in CT_3^o \ if \ cr(ct_i \oplus ct_3) <=$ $<= cr \ \& \ cr(ct_i \oplus ct_3) <=$ $<= cr(ct_i \oplus ct_1) \& cr(ct_i \oplus ct_3) <=$ $<= cr(ct_1 \oplus ct_2)$ де $ct_i \mid i = \overline{1,N}$ – реалізації БНМ $\left\|ct_i^{(j)}\right\|$.
	8.4	Виконується корекція КОН для кластерів $\{CT_m^o \mid m = \overline{1,2}\}$:	Реалізації $\{ss^{(j)}, j = \overline{1,n}\}$, що потрапили до КОН категорії CT_3^o, вилучаються з контейнеру $\{CT_m^o\}$. Радіус КОН $\{CT_m^o\}$ розраховується так: $cr_m[\xi] := cr_m[\xi] - 1.$
	8.5	Обчислення поточного показника ІУФР:	$\overline{CE}^* = (1/M) \cdot \sum_{m=1}^{M} \max_{\{ls\}} CE_c.$
	8.6	Формування множини $\{ct_m\}$ еталонних реалізацій $\{CT_m^o[\xi]\}$:	Правило для визначення координат: $ct_{m,i} = \begin{cases} 1, & if \ \frac{1}{n}\sum_{j=1}^{n} cr_{m,i}^{(j)} > 1/2; \\ 0, & else. \end{cases}$
	8.7	Перевірка умов:	$\begin{cases} if \ cr_3[\xi] < cr(ct_1 \oplus ct_3) \& cr_3[\xi] < \\ < cr(ct_2 \oplus ct_3) \ then \to 8.8; \\ else \ 8.9. \end{cases}$
	8.8	Збільшення радіусу:	$cr_3[\xi] := cr_3[\xi] + 1.$
	8.9	Розраховується *optim* радіус КОН кластера CT_3^o:	За умов: $cr_3[\xi] < cr(ct_1 \oplus ct_3) \&$ $cr_3[\xi] < cr(ct_2 \oplus ct_3).$
9	9	Перевірка умов:	$\begin{cases} if \quad ca[l] \le 0{,}5 \cdot ca_{low} \ then \to 2 \\ else \ 10. \end{cases}$ ca_{low_i} – перевірочні допустимі відхилення для ознак ОР, які визначаються на основі методів мультифрактального аналізу, показника Херста, рухомого вікна [132, 190, 145, 200, 210 та ін.].

продовження таблиці 3.1

10	10	Перевірка умов:	$\begin{cases} if \quad \overline{CE}[l] \notin IS_{CE} \ then \to 11 \\ else \ 2. \end{cases}$
11	11	Виконується процедура пошук глобального максимального значення $\overline{CE}$ в робочому діапазоні ознак ОР:	$ca^* = \underset{IS_{ca}}{\operatorname{argmax}}\{\max_{IS_{CE} \cap IS_{cr}} \overline{CE}\} \&$ $\overline{CE}^*[l] := extremCE_m[l].$
12	12	На основі методів мультифрактального аналізу, показника Херста, рухомого вікна та ін. визначається оптимальний параметр поля ca ознак ОР для КОН:	$A^{op}_{low_i} = lm_i - ca^{op} \frac{ca_{low_i}}{100};$ $A^{op}_{up_i} = lm_i + ca^{op} \frac{ca_{up_i}}{100}.$
13	13	Процедура розбиття БПОЗ ОР на чотири кластери $\{CT^o_m[\xi] \mid m = \overline{1,4}\}$:	
	13.1	Визначається бінарна матриця кластера $\{CT^o_4\}$:	За умов: $cr(ct_1 \oplus ct_4) \to \min,$ $cr(ct_2 \oplus ct_4) \to \min \&$ $cr(ct_3 \oplus ct_4) \to \min,$ де ct_1, ct_2, ct_3 – еталонні реалізації кластерів $\{CT^o_m \mid m = \overline{1,3}\}$, відновлених при виконанні етапу 8.
	13.2	Значення радіусів кластеру CT^o_4 виставляється рівним "0":	$cr_4[\xi] := 0.$
	13.3	Визначаємо реалізації ОР, які потрапили в кластер CT^o_4:	Правило: $ct_i \in CT^o_4, if \ cr(ct_i \oplus ct_4) <= cr_4[\xi],$ де $ct_i \mid i = \overline{1, N_4}$ – реалізації БНМ $\left\| ct_i^{(j)} \right\|$.
	13.4	Обчислення поточного показника ІУФР:	$\overline{CE}^* = (1 \Big/ M) \cdot \sum_{m=1}^{M} \max_{\{ls\}} CE_c.$

продовження таблиці 3.1

	13.5	Формування множини $\{ct_m\}$ еталонних реалізацій для кластерів $\{CT_m^o[\xi]\}$:	Правило для визначення координат: $ct_{m,i} = \begin{cases} 1, & if \ \frac{1}{n}\sum_{j=1}^{n} cr_{m,i}^{(j)} > 1/2; \\ 0, & else. \end{cases}$
	13.6	Додається наступна ознака в кластері CT_4^o:	$ct_4 := ct_4 + 1.$
	13.7	Визначаємо оптимальний радіус КОН кластера CT_4^o:	За умов: $cr_4[\xi] < cr(ct_1 \oplus ct_4),$ $cr_4[\xi] < cr(ct_2 \oplus ct_4),$ $cr_4[\xi] < cr(ct_3 \oplus ct_4).$
14	14	Зупинка роботи.	

В якості критерію оптимізації параметрів, під час навчання ЕС у складі СІРКЗ, використовувалися статистичні параметри (інформаційні міри) для варіантів рішень з двома альтернативами, відповідно вирази (2.17), (2.24) та (2.25) для модифікованого ентропійного показника, а також міру Кульбака – Лейблера (для трьох гіпотез) [26, 50-52, 80, 94, 137, 148, 151].

Додатково розглянемо алгоритм який дозволяє виконувати паралельну оптимізацію при обчислення контрольних допусків в ході аналізу ЕС реалізацій ознак при розпізнавані ОР [26, 50-52, 80, 94, 137, 148, 151, 209].

Такий підхід, коли виконується одночасна оптимізація ПДВ – ($\{ca_{K,i}\}$), дозволяє на кожному кроці навчання змінювати ПДВ для всіх ознак одночасно. Алгоритм дозволяє в ході навчання відновлювати оптимальні параметри КОН для класів розпізнавання CT_m^o.

Для наочності етапи розбиття ПОЗ ОР на кластери подамо у табличному вигляді, див. таблиця 3.2.

Таблиця 3.2

Етапи алгоритму оптимізації ПДВ на реалізацій ознаки розпізнавання кібератак

№	Етап	Дія	Математичний опис для алгоритму кластеризації реалізацій ознак ОР
1	1	Значення лічильника кроків зміни ПДВ ca_i на реалізації ознаки ОР виставляється рівним "0" :	$l := 0$.
2	2	Обчислення нижнього $A_{low_i}[l]$ та верхнього $A_{up_i}[l]$ ПДВ ознак ОР для всього простору RS :	$A_{low_i}[l] = lm_{1,i} - ca\frac{ca_{low_I}}{100};$ $A_{up_i}[l] = lm_{1,i} + ca\frac{ca_{low_I}}{100},$ де $lm_{1,i}$ – i-та ознака вектору-еталону реалізації lm_1 для базового класу CT_1^o. (Приймаємо, що CT_1^o характеризує найбільш прийнятні для ІБ стан АЕС). $l := l + 1.$
3	3	Формування (БНМ) $\|ct_i^{(j)}\|$:	Правило: $ct_{m,i}^{(j)} = \begin{cases} 1, if\ A_{low_I}[l] < lm_i^{(j)} < A_{up_i}[l]; \\ 0, else. \end{cases}$
4	4	Формування множини $\{ct_m\}$ для векторів еталонів реалізацій класів CT_m^o ОР:	$ct_{m,i} = \begin{cases} 1, if\ 1/n \sum_{j=1}^{n} ct_{m,i}^{(j)} > 1/2; \\ 0, else, \end{cases}$ де n – кількість реалізацій ОР (ознак), які належать кластеру відповідного класу CT_m^o.
5	5	Реалізація процедури розбиття множини $\{ct_m\}$ на пари найближчих сусідніх векторів-еталонів:	Використовуються, зокрема, методики [7, 9, 28, 50-52, 61, 80, 94, 189, 210]

продовження таблиці 3.2

6	6	Реалізація процедури відновлення КОН для відповідного класу CT_m^o:	
	6.1	Значення лічильника класів розпізнавання виставляється рівним "0":	$m := 0$
	6.2	Збільшення значення лічильника:	$m := m + 1$
	6.3	Значення лічильника кроків зміни радіуса КОН виставляється рівним "0":	$cr := 0$
	6.4	Збільшення значення лічильника:	$cr := cr + 1$
	6.5	Обчислення поточного показника ІУФР:	$\overline{CE}^* = (1/M) \cdot \sum_{m=1}^{M} \max_{\{ls\}} CE_c .$
	6.6	Перевірка умови:	$\begin{cases} if\ CE_m \notin IS_{CE} \quad then \to 6.4 \\ else\ 6.7. \end{cases}$
	6.7	Обчислення поточного показника ІУФР:	$\overline{CE}^* = (1/M) \cdot \sum_{m=1}^{M} \max_{\{ls\}} CE_c .$
	6.8	Обчислення глобального максимуму показника ІУФР:	$CE_m^*[l] := \underset{\{cr\}}{extrem} CE_m[l, cr]$
	6.9	Обчислення оптимального радіусу КОН класу ОР CT_m^o:	$cr_m^*[l] := \arg \underset{\{cr\}}{extrem} CE_m[l, cr]$
7	7	Перевірка умови:	$\begin{cases} if\ m \notin M \quad then \to 6.2 \\ else\ 8. \end{cases}$
8	8	Обчислення усередненого значення показника ІУФР:	$\overline{CE}_{cp} = (1/M) \cdot \sum_{m=1}^{M} \max_{\{ls\}} CE_c .$

продовження таблиці 3.2

9	9	Перевірка умови:	$\begin{cases} if\ ca[l] \le ca_{low} / 2\ then \to 2 \\ else\ 10. \end{cases}$
10	10	Перевірка умови:	$\begin{cases} if\ \overline{CE} \notin IS_{CE}\ then \to 11 \\ else\ 6.8\ \&\ 6.9. \end{cases}$
11	11	Обчислення глобального максимуму ІУФР в допустимому діапазоні визначення його функції:	$ca^* = \arg\max_{IS_{ca}} \{ \max_{IS_{CE} \cap IS_{cr}} \overline{CE} \}.$
12	12	Зупинка роботи алгоритму.	

Структурно-блочну схему алгоритму, представленого у таблиці 3.2, подано на рис. 3.1. Вхідними даними є масив навчальних вибірок, отриманих на підставі даних таблиці 2.4:

$$LM[kl][implementa\ tion][j], \tag{3.13}$$

де kl – номер навчальної матриці для класу ОР;

$implementation$ – номер реалізації в навчальній матриці;

j – номер реалізацій ознаки розпізнавання для ОР.

Розглянемо призначення функціональних модулів у структурній схемі алгоритму навчання ЕС у складі СІРКЗ для запропонованих етапів паралельної оптимізації ПДВ для ознак розпізнавання, див. рис. 3.1. Для наочності етапи подамо у табличному вигляді, див. таблиця 3.3.

Таблиця 3.3

Призначення модулів структурної схеми навчання ЕС у складі СІРКЗ для запропонованих етапів паралельної оптимізацією ПДВ для ознак розпізнавання

Модуль	Пояснення
1	Старт.
2	Формування масивів навчальних вибірок.
3	Начальне значення тимчасової змінної *optim_medium_ІУФР*, яка використовується для прямого пошуку глобального максиму ІУФР навчання ЕС для ідентифікації ОР виставляється рівним «0».
4	Ініціалізація лічильника циклів зміни параметру ca (діапазон контрольних відхилень у ББНМ).
5	Формування на кожному кроці навчання (див. табл. 3.1 етап 2) масивів поточних верхніх та нижніх ПДВ для реалізацій ознак розпізнавання (A_{low} & $A_{up}(0, ca)$).
6	Метод навчання ЕС у складі СІРКЗ відповідно до таблиць етапів, див. табл. 3.1 та 3.2.
7	Перевірка умов результативності навчання if (CE_optim[ca] > optim_medium_ІУФР).
8, 9	Пошук глобального максимуму ІУФР навчання ЕС.
10	Зупинка.

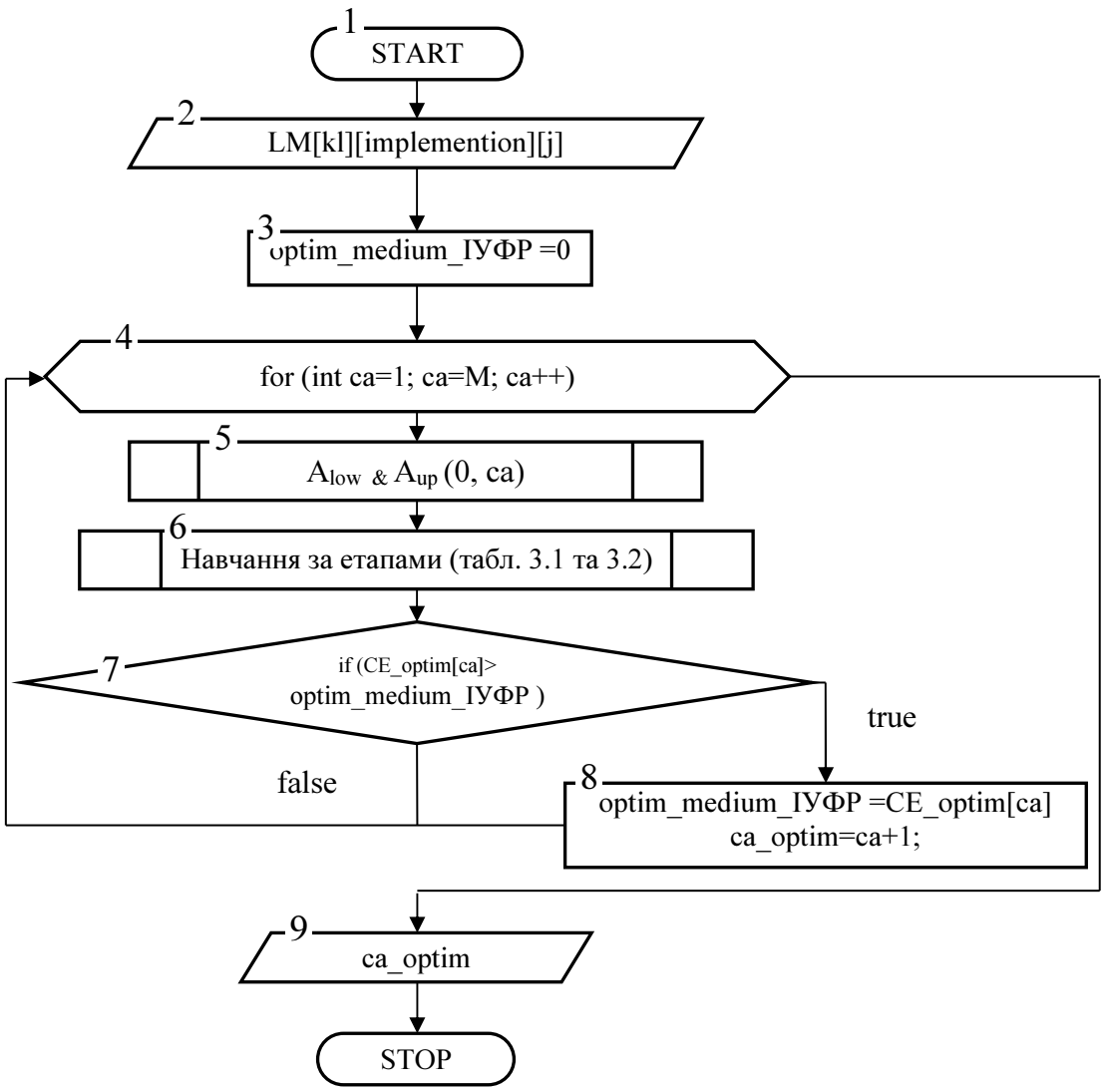


Рис. 3.1. Структурна схема алгоритму пошук у глобального максимуму ІУФР навчання ЕС

Для оцінки оптимальності багатокритеріальних систем, зокрема й ЕС у складі СІРКЗ, застосуємо метод Парето [43, 62, 70, 80]. Оптимізація пов'язана з пошуком керуючих впливів в ході навчання ЕС, що забезпечують оптимальні значення ІУФР. Остаточний вибір варіанта рішення на початковому етапі перевіряється експертами з ІБ, а також, враховує множину можливих станів КВІС та АСР (СІРКЗ) – $SS = \{SS_1, SS_2, ..., SS_d\}$. Стани АСР визначені нечітким числом на шкалі оцінок (0, 1).

Ступінь приналежності найкращого, з точки зору ЕС або експерта, варіанта Парето–оптимального рішення щодо стратегій, які обрані, визначається за формулою:

$$\max\left[\sum_{j=1}^{h}\sum_{l=1}^{d}\widetilde{z}_{ij}\otimes\widetilde{p}_{j}\otimes\widetilde{p}_{l}^{SS}\right]=\max_{W_i\in W}CE(W_i(x)), \tag{3.14}$$

де $\otimes$ – триангулярна норма (Т–норма), яку ми розглядаємо, як логічний добуток по Заде [183, 190, 209];

$W_i(x)$ – остаточний вибір варіанта рішення ЕС (або експерта);

$\widetilde{z}_{ij}$ – нечітка оцінка корисності *i*-го варіанта вирішення завдання розпізнавання ОР ЕС, яка визначається значенням ІУФР;

$\widetilde{p}_j$ – оцінка станів КВІС в процесі розпізнавання ОР;

$\widetilde{p}_l^{SS}$ – оцінки станів ЕС в процесі розпізнавання кібератак.

Визначення ступеня приналежності найкращого варіанту Парето–оптимального нечіткого рішення для формування БЗ для ЕС виконано із застосуванням модифікованого критерію Вальда і критерію Севіджа [45, 88].

Для формування переліку можливих змін станів КВІС (КВКС) застосуємо наступний терм:

$$SS_R=\left\langle EUM^{*}, SDN, RDN, ADN, MIF, IR\right\rangle, \tag{3.15}$$

де EUM^{*} – множина сутностей КВІС або КВКС (наприклад, підмножини вузлів КВІС або КВКС які мають um^{*} – потенційні уразливості);

SDN – множина суб'єктів КВІС або КВКС;

RDN – множина ребер графа станів КВІС SS_R (наприклад, ребра графа, що описує права доступу користувачів до EUM^{*});

ADN – множина ребер графа станів КВІС SS_R, що відповідають отриманому доступу до EUM^*;

MIF – множина ребер графа станів КВІС SS_R, що відповідають інформаційним потокам між EUM^* ($um^* \subset EUM^*$);

IR – функція ієрархії EUM^*.

Приймаємо, що в кожному стані КВІС (SS_R) повинні виконуватися наступні умови [9, 28, 50-52, 61, 80, 88, 93, 94, 101, 189, 210]:

1) для кожного вузла $um^* \subset EUM^*$ визначаються довірені користувачі, що володіють правом доступу володіння до кожної сутності (наприклад, інформаційному масиву – $M_{k\,inf}$);

2) множини EUM^*, SDN й функція IR не змінюються на всіх траєкторіях графа станів системи.

Для одержання зловмисником (суб'єктом-порушником) SDN_x права володіння щодо суб'єкта SDN_i йому необхідно одержати доступ не тільки до сутності EUM_z^*, але й доступ на запис/читання до деякої сутності $eum_l \in EUM^*$, що є інтерфейсом або портом деякого суб'єкта-процесу $pro \in SDN$, що здійснює надання прав доступу SDN_i на основі даних у сутності EUM^*. При цьому сутності EUM_z^* і eum_l є асоційованими із суб'єктом pro_r^m; сутності eum_l й pro_r^m, як правило, розміщені на одному вузлі мережі, а сутності EUM_z^* й EUM_y^* можуть бути розміщені на різних вузлах КС.

Фактори, що впливають на вибір рішення щодо формування вирішального правила RUL_i (див. вираз 2.34), представлені у вигляді лінгвістичних змінних у БЗ ЕС (див. рис. 2.6 та табл. 3.4), для яких вибрані універсальні множини та терми.

Для формалізації лінгвістичних змінних застосовується модель (3.14) Парето–оптимального рішення для функції належності, що зменшує розмірність задачі підбору цих параметрів при навчанні [9, 28, 50-52, 61, 80, 88, 93, 94, 101, 107, 123, 132, 137, 189, 210].

Для класів DoS/DDoS, Probe, R2L, U2R та вірусного зараження АРМ у складі КВІС (КВКС) алфавіт класів розпізнавання складався із трьох класів, див. табл. 3.4.

Тоді вирішальне (розв'язувальне) правило RUL_i, яке описує стани КВІС (або КВІС), можна представити в такому вигляді (див. табл. 3.5).

Для подальшого вирішення завдань дисертаційної роботи та з метою перевірки отриманих у підрозділах 2.1, 2.2 та 3.1 результатів, потрібно провести дослідження за допомогою імітаційного моделювання для моделей та алгоритмів кластерізації реалізацій ознак ОР, які ґрунтуються на ентропійному та інформаційно-дистанційному критерії Кульбака – Лейблера, а також алгоритму оптимізації контрольних допусків на ознаки розпізнавання кібератак. Дане завдання вирішується у розділі 3.2.

Таблиця 3.4

База знань ЕС для ОР у системі ІБ КВІС (КВКС)

Класи загроз КВІС (КВКС)		Атрибути		Ознаки $\{rc_1,\ldots,rc_n\}$	Класи розпізнавання для процедури навчання ЕС (гіпотези)	Інформативність значення ознаки IZ_{rc} [80]	Універсум	Терми для лінгвістичної оцінки
Можливі загрози ІБ ІКСТГ	**Відомі загрози**	CT_1	Відмова в обслуговуванні елементів КВІС, КВКС, або АСК, SCADA, HMI, PLC та ін.	1–не працюють штатні компоненти; 2 – не працюють штатні компоненти; 3–ін.	4) Основна робоча гіпотеза яка (базова) – hy_{γ_1}: ознака (ознаки) rc_i ОР (*RS*) та показник *IE* (див. вираз 2.27) знаходиться у межах звичайного стану КВІС;	$0 \le IZ_{rc} \le 1$	[0,1], у. о.	некритичний (нкр), критичний (кр)
		CT_2	Викрадення інформації або компонентів КВІС, КВКС, SCADA, HMI, PLC	1– об'єктивні ознаки (ОО) (наприклад, поява конфіденційної інформації у ЗМІ); 2–суб'єктивні ознаки (СО); 3–ін.			[0,1], у. о.	виявлені (в), частково невиявлені (чв), невиявлені (нв)
		CT_3	Привласнення особистості у КВІС, КВКС або АСК, SCADA, HMI, PLC	1– ОО (наприклад, зафіксовані спроби роботи під чужим логіном); 2–СО; 3–ін.	5) гіпотеза hy_{γ_2} – ознака (або ознаки) rc_i ОР (*RS*) та показник *IE* дозволяють зробити висновок, що значення показника *IE* є меншими на норму;		[0,1], у. о.	(в), (чв), (нв)
		CT_4	Модифікація інформації у КВІС, КВКС або АСК, SCADA, HMI, PLC	1–зміна контенту; 2–зміна структури документів; 3-ін.			[0,1], у. о.	(в), (чв), (нв)
		CT_5	Злам пароля користувача	1 – виявлення кейлоггера; 2–атипова поведінка користувача; 3–ін.			$[0, N_a]$	зафіксовані СЗІ (зф), незафіксовані СЗІ(нф)

продовження таблиці 3.4

Можливі загрози ІБ ІКСТГ	**Відомі загрози**	CT_6	Вірусна атака	1–незвичайні прояви в роботі ІС; 2–зміни заданої в передостанньому сеансі роботи з ЕОМ структури файлової системи; 3-ін.	6) гіпотеза hy_{γ_3} – показник *IE* дозволяють зробити висновок, що значення показника *IE* є більшими за норму;	$0 \le IZ_{rc} \le \le 1$	[0,1], у. о.	(в), (чв), (нв)
		CT_7	Пошук залишкової інформації у КВІС	1– ОЗ; 2–СО; 3–ін.			[0,1], у. о.	(в), (чв), (нв)
		CT_8	Несанкціонований запуск ПЗ КВІС	1–незвичайні прояви в роботі ПЗ ІС; 2– атипова поведінка ПЗ; 3–ін.			$[0, N_a]$	(зф), (нф)
		CT_9	Зміна конфігурації СЗІ КВІС	1–чужорідне тіло; 2–сторонні сигнали; 3-ін.			$[0, N_a]$	(зф), (нф)
		CT_{10}	Несанкціоноване знищення даних КВІС	1– ОО; 2–СО; 3–ін.			[0,1], у. о.	(нкр), (кр)
		CT_{11}	Несанкціоноване відкриття файлів КВІС	1– ОО; 2–СО; 3–ін.			[0,1], у. о.	(в), (чв), (нв)
		CT_{12}	Зміна конфігурацій ПЗ КВІС	1–нові модулі ПЗ; 2–стороннє ПЗ; 3–ін.			[0,1], у. о.	(в), (чв), (нв)
		CT_{13}	Зміна конфігурації обладнання КВІС	1–чужорідне тіло; 2–сторонні сигнали; 3–ін.			[0,1], у. о.	(в), (чв), (нв)
		CT_{14}	Зміна конфігурації обладнання SCADA для КВКС	1–чужорідне тіло; 2–сторонні сигнали; 3–сліди установки; 4–ін.			[0,1], у. о.	(в), (чв), (нв)

		CT_{15}	Зміна конфігурації обладнання систем супутникової навігації КВКС	1–Рівень сигналу; 2–Однаковий рівень сигналу від різних супутників; 3–Візуальні ознаки відхилення від маршруту; 4–Ін.	Гіпотези: $hy_{\gamma 1}$; $hy_{\gamma 2}$; $hy_{\gamma 3}$.	$0 \le IZ_{rc} \le 1$	[0,1], у. о.	(в), (чв), (нв)
		CT_{16}	Зміна конфігурації обладнання систем оповіщення та відеоспостереження	1–чужорідне тіло; 2–сторонні сигнали; 3–сліди установки; 4–ін.			[0,1], у. о.	(в), (чв), (нв)
		CT_{17}	Порушення доступності ІМ та ПЗ КВІС або КВКС	1–не працюють штатні компоненти ІС ТГ (ІМ та ПЗ); 2–ін.			[0,1], у. о.	(нкр), (кр)
		Інші класи атак за [9, 28, 50, 61, 80, 88, 93, 94, 101, 189, 210]						
Неописані атаки		CTN_{31} – CTN_{3m}	Потенційні класи атак для яких немає описаних атрибутів					

Базові стани КВІС, методи та засоби протидії кіберзагрозам	
Базові стани системи описані ББНМ	**Методи протидії атакам у КВІС**
SS_1 – у КВІС встановлене сертифіковане ПЗ та оновлення до нього; SS_2 – в системі присутні мережеві сервіси; SS_3 – система підтримує багатозадачність; SS_4 – підтримка багатокористувацького режиму; SS_5 – встановлені пристрої введення / виводу; SS_6 – наявність пристроїв «гарячої заміни»; SS_7 – наявність зовнішніх каналів зв'язку; SS_8 – наявність «уразливих» систем з'єднаних із КВІС; SS_9 – наявність сертифікованих ЗЗІ; SS_i - інше.	D_1 – ідентифікація і аутентифікація; D_2 – блокування безконтрольного доступу; D_3 – захист від вірусів; D_4 – контроль цілісності даних; D_5 – знищення залишкових даних; D_6 – захист ПЗ та ІМ від дослідження; D_7 – резервування інформації; D_8 – відновлення і самовідновлення компонентів КВІС; D_9 – перевірка сертифіката безпеки; D_{10} – блокування запуску ПЗ; D_{11} – криптографічний захист у КВІС; D_{12} – ін.

Таблиця 3.5

Вирішальне правило RUL_i для визначення стану КВІС (КВКС) у випадку розпізнавання загрози для ІБ або виявлення кібератаки

Правило	Вихідний стан КВІС SS_R	Результуючий стан КВІС SS'_R
$RUL = (SDN_x, SDN_y, EUM^*_z, eum_l, pro^m_r)$	SDN_x, SDN_y, $pro^m_r \in EUM^*$, eum_l, $EUM^*_z \in EUM$, $eum_l \in pro^m_r$, $(SDN_x, eum_l, write_r / read_r \in RDN)$, $EUM^*_z \in$ SDN y і або $SDN_x = SDN_y$, або $(EUM^*_z, SDN_x, write_m / read_m) \in MIF, CT^0_m$, $MC \in AL(CT^0_m)$ де AL – алгоритми які використовуються для розпізнавання, зокрема, й метод розбиття простору ознак на кластери в ході реалізації процедури розпізнавання кібератак (табл. 3.1); підмножина $MC^{AL}(CT)$	$SS_R = SS'_R$, $EUM^* = EUM'^*$, $ADN = ADN'$, $IR = IR'$, $MIF = MIF'$, $RDN' = RDN'(SDN_x, SDN_y)$, $CT \in (CT_1,...CT_l)$, $MC \in AL$ $(CT \in (CT_1,...CT_l))$

3.2. Імітаційне моделювання адаптивної системи інтелектуального розпізнавання із використанням процедури нечіткої кластеризації та паралельної оптимізації контрольних відхилень для реалізацій ознак кібератак

Великі системи, у тому числі КВІС (КВКС), оснащені комплексами ЗЗІ, зокрема й СІРКЗ, складаються з сотень, а в деяких випадках із тисяч елементів і ще більшої кількості зв'язків між ними. Такі системи характеризуються неоднорідністю елементів і неоднорідністю зв'язків. Незважаючи на те, що окремі елементи або зв'язки прекрасно описуються моделями дискретної математики [57, 83] або теорії масового обслуговування [15, 22, 78], про систему в цілому цього сказати не можна. Природною альтернативою є використання імітаційного моделювання, яке дозволяє поєднати між собою різнорідні математичні моделі елементів, що входять до складу КВІС (КВКС). Імітаційне моделювання є одним із методів, які дозволяють оцінити ЗЗІ ІСТГ та її реакцію на спроби НСД (збурення) за рядом показників [9, 27, 61, 79, 88, 93, 101, 189, 201, 210].

При імітаційному моделюванні доцільне використання схематичних моделей, які за своєю суттю є відображенням дійсного перебігу подій, що сприяє поглибленому розумінню процесу функціонування системи. Схема етапів створення імітаційних моделей компонентів ЗЗІ для КВІС (КВКС) та взаємозв’язки валідації, верифікації і встановлення довіри представлено на рис. 3.2.

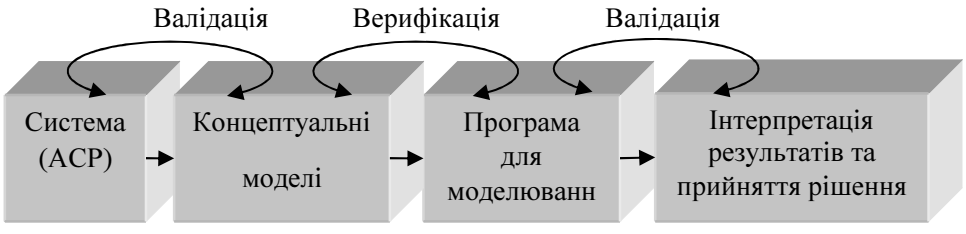


Рис. 3.2. Схема етапів створення імітаційний моделей компонентів АСР

Валідацію можна протиставити етапу інтерпретації або аналізу вихідних даних моделювання, який становить статистичну задачу, що пов'язана з оцінкою достовірності результатів, отриманих за допомогою імітаційної моделі АСР для КВІС (КВКС).

Кінцева перевірка адекватності імітаційної моделі підтверджується за умови, що її вихідні дані ідентичні вихідним даним реальної АСР. Якщо система, аналогічна запровадженій, існує у поточний момент часу, то розробляють імітаційну модель і порівнюють вихідні дані. У тому випадку, коли два комплекти даних виявляються подібними, модель системи рахується адекватною.

За допомогою імітаційного моделювання при створенні ЗЗІ можуть вирішуватися такі завдання: визначення шляхів удосконалення СІРКЗ для КВІС (КВКС) на підставі аналізу різних варіантів технічної, технологічної, а також організаційної перебудови та дослідження наслідків прийнятих рішень. Імітаційне моделювання дозволяє відпрацьовувати не тільки різні варіанти структур і режимів функціонування технічних засобів і програмного забезпечення, але і різних форм функціонування ЗЗІ ІКСТГ.

Для імітаційного моделювання роботи АСР був обраний пакет Simulink [79, 100].

Всі моделі та методи, описані в розділах 2.1, 2.2 та 3.1 були реалізовані в пакеті MATLAB 7/2009 для подальшого дослідження режимів роботи АСР для КВІС в умовах протидії різним варіантам кібератак, а також для перевірки адекватності отриманих моделей.

Імітаційна модель (сегмент КВІС із АСР) складається з однієї лінії передачі даних і трьох станцій (автоматизованих робочих місць – АРМ), рис. 3.3.

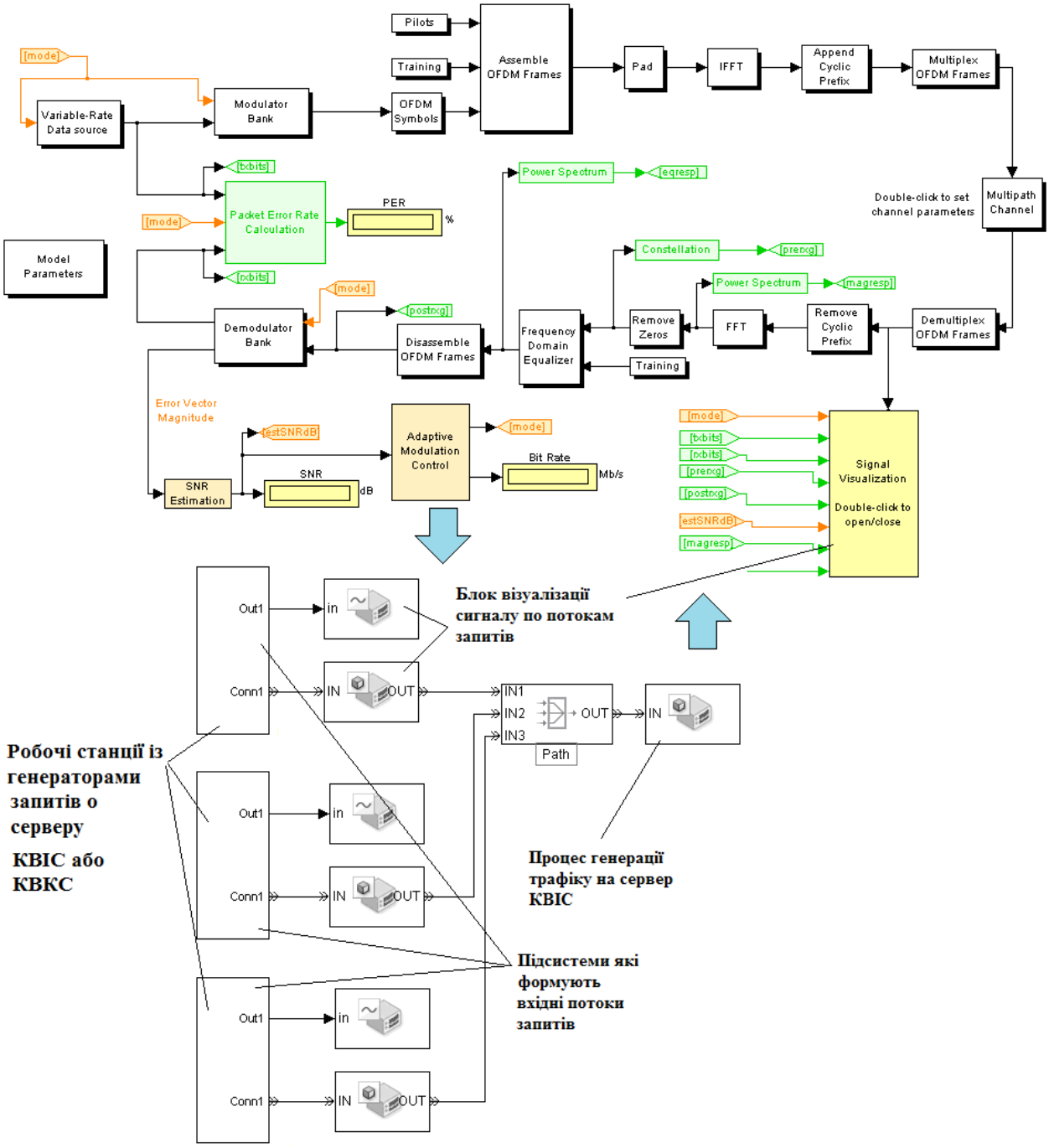


Рис. 3.3. Схема імітаційного моделювання АСР у сегменті КВІС

У блоці аналізу трафіку (кібератак за ознаками), використовуючи блок розпізнавання [79] та закладені вирішальна правила RUL_i, можна блокувати відповідні атаки та несанкціоновану мережеву активність.

Для реалізації процесу розпізнавання по окремих класах ОР у імітаційній моделі АСР за допомогою пакету розширення Fuzzy Logic

Toolbox були складені відповідні правила для системи розпізнавання, рис. 3.4.

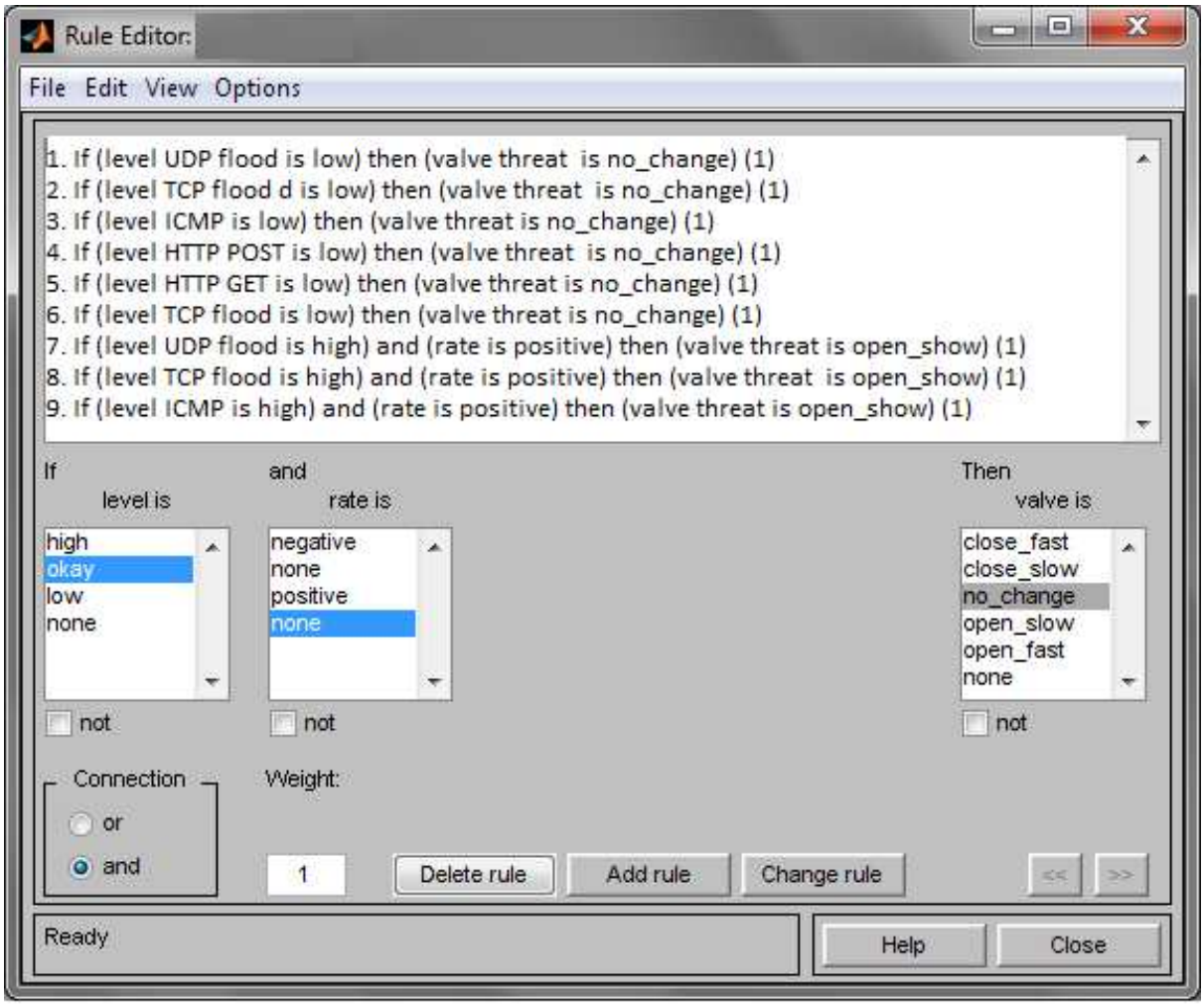


Рис. 3.4. Система правил для ЕС у складі СІРКЗ

Для візуалізації сигналів був спроектований спеціальний блок – «Signal Visualization», рис. 3.5, який дозволяє аналізувати основні параметри ЛОМ на рівні переданих пакетів даних [101, 133].

За допомогою генератора трафіка створювалися атаки які за ознаками [43, 86, 96, 201, 210], належать класам – Probe, U2R, R2L, Dos/DDoS.

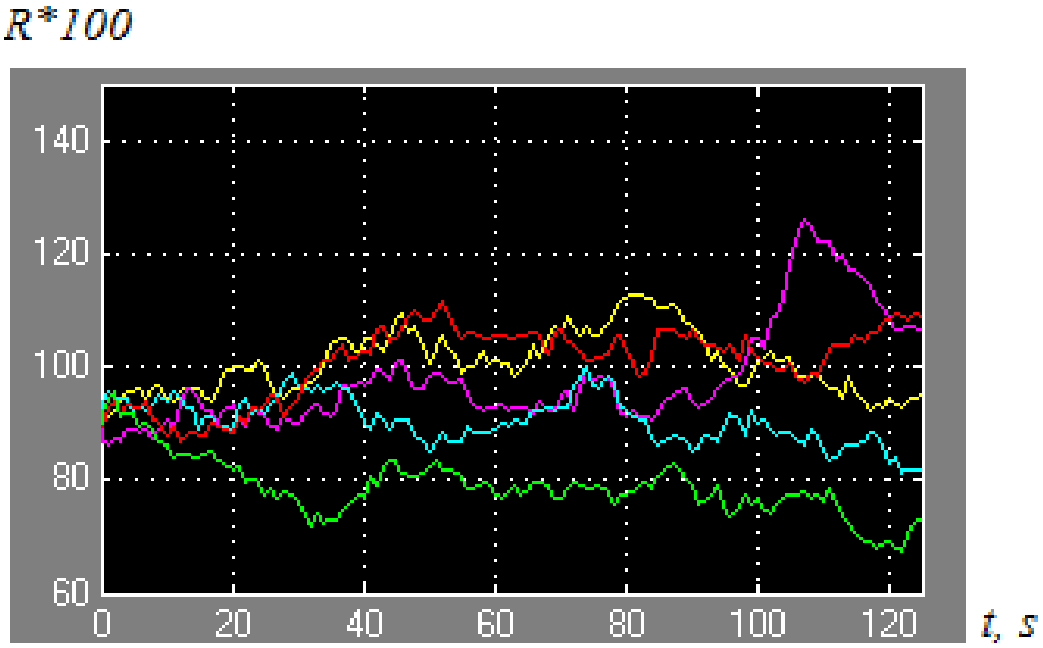


Рис. 3.5. Блок «Signal Visualization» для візуалізації сигналів у АСР

Лістинги основних функцій, що описують сегмент ЛОМ КВІС, у вигляді *m* файлу наведені в додатку В.

У блоці аналізу трафіку, використовуючи вирішальні правила для АСР (розділ 2), можна будувати розподілені системи виявлення та блокування складних кібератак, несанкціонованої мережевої активності, а також інших завдань інформаційної безпеки КВІС.

Модуль формування бінарних векторів розпізнавання (МФББР), дозволяє отримати наочне уявлення про процеси на виході кожної станції і на загальній лінії, а також, визначити важливі характеристики АСР. Наприклад, на рис. 3.6, зображений тестовий трафік на лінії, коли після процедури кластеризації реалізацій ознак на виході, виконується бінеарізація векторів-реалізацій класів ОР (кібератак), шляхом порівняння поточних реалізацій ознак з їх відповідними перевірочними допусками $\{ca_{K,i}\}$, які містяться у БД АСР. Відповідно, на рис. 3.6 показаний модуль детектора для виявлення аномальності квантового сигналу під час процедури розпізнавання атаки.

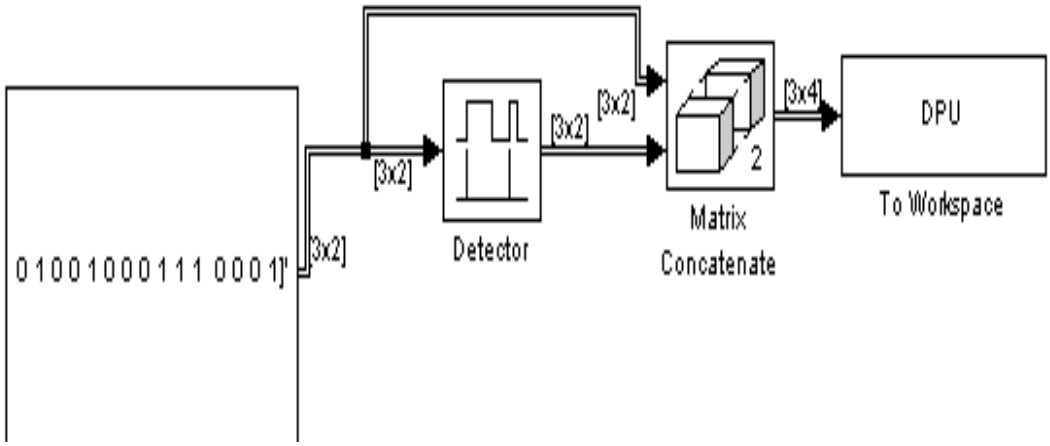


Рис. 3.6. Фрагмент тестової послідовності на лінії для об'єктів, що перевіряються, належність ОР до відповідного класу за бінарними ознаками

Згідно з викладками, наведеними в розділі 2, на рис. 3.8–3.11 подано схеми модулів зміни параметрів мережевого трафіка та продуктивності ПК для різних варіантів кібератак (DoS/DDoS, R2L, U2R та випадків зараження ПК вірусами). Наведені схеми використовувалися для побудови ОВН та МІЕТ для ЕС «Analyzer of cyberthreats» у складі СІРКЗ.

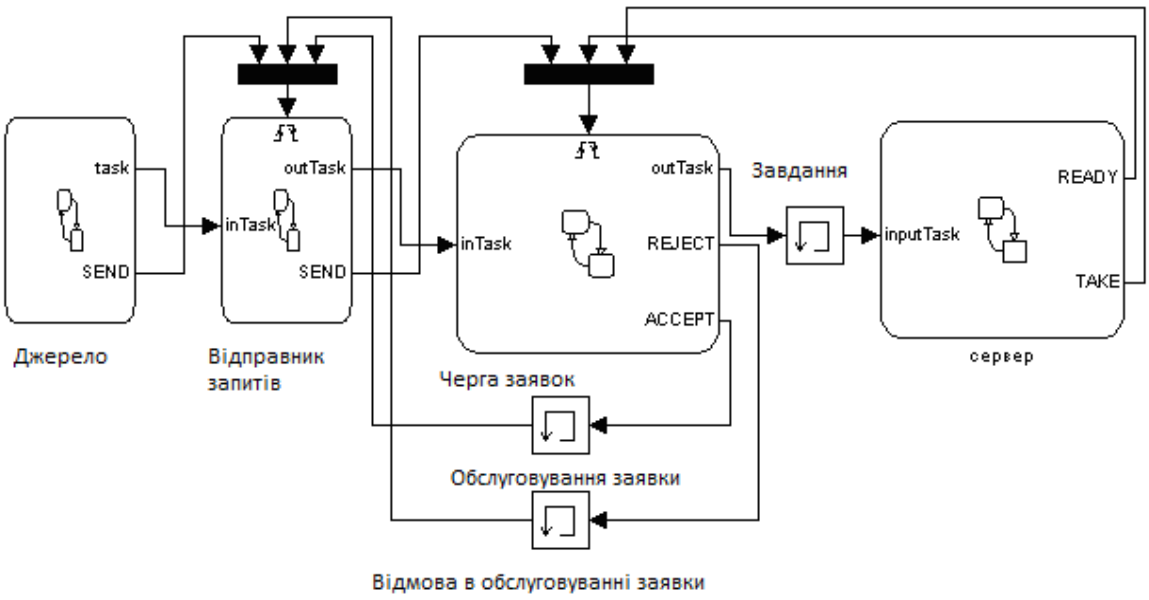


Рис. 3.7. Схема функціонування сервера КВІС для DoS/DDoS атак

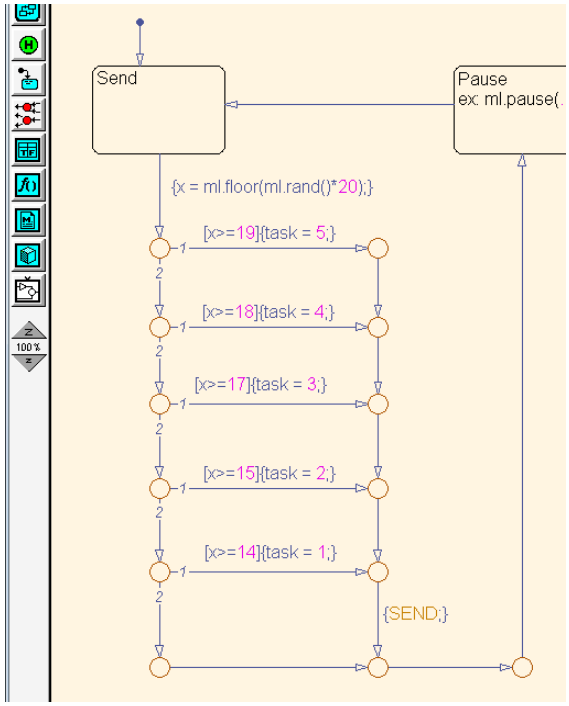


Рис. 3.8. Модуль «Джерела запитів»

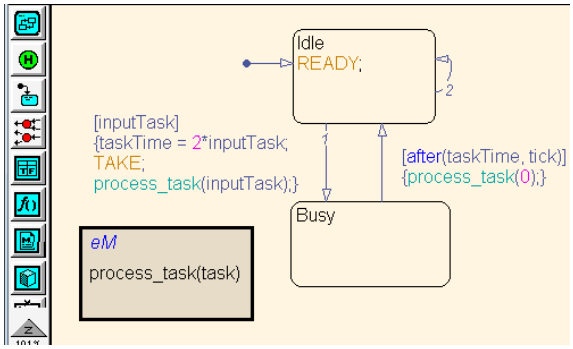


Рис. 3.9. Модуль «Сервер»

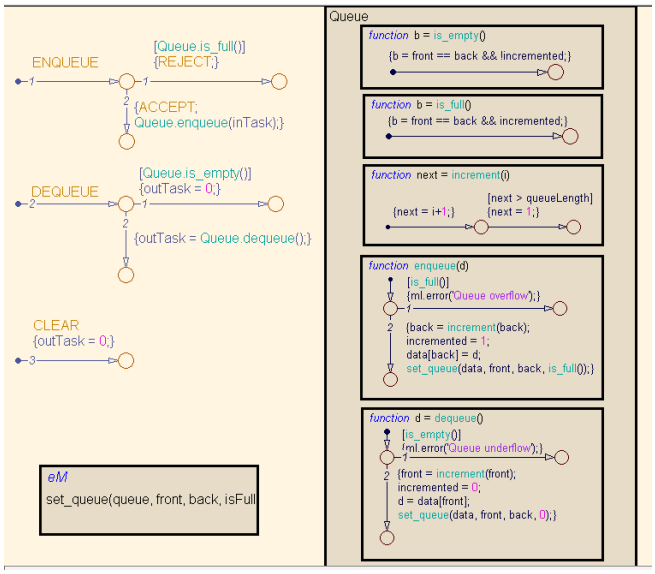


Рис. 3.10. Модуль «Відправник запитів»

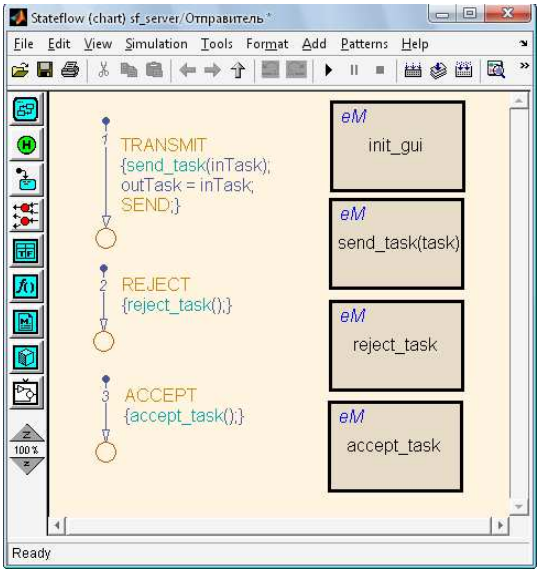


Рис. 3.11. Модуль «Черга запитів»

Наприклад, розглянемо застосування алгоритму пошук у глобального максимуму ІУФР навчання ЕС (див. табл. 3.1–3.3 та рис. 3.1) та процедури паралельної оптимізації ПДВ для реалізацій ознак розпізнавання мережевих атак класів DoS/DDoS, Probe, R2L, U2R та вірусного зараження АРМ у складі КВІС (КВКС), що дозволить фахівцю з ІБ обирати більш ефективні стратегії реагування на потенційні загрози, існуючі аномалії або виявлену кібератаку.

У відповідності до рекомендацій [80, 113, 137, 180, 210], ББНМ або ОВН для розглянутих у таблиці класів ОР, мали від 50 до 65 реалізацій. Для класів мережевих атак [86] (DoS/DDoS, Probe, R2L, U2R) кількість ознак розпізнавання становить 12 [80] – 41[43, 86] для вірусних атак 7–15 [86, 110] ознак. Для кожного класу вектор-реалізація для алгоритму, показаному на рис. 3.2, подається у вигляді структурованої послідовності ознак розпізнавання. Відповідні бінарні ознаки прийнято за результатами попередніх тестових вибірок отриманих за допомогою програми Wireshark , див. рис. 3.12 та даних досліджень наведених у [9, 28, 50, 61, 80, 88, 93, 94, 132, 189 та ін.].

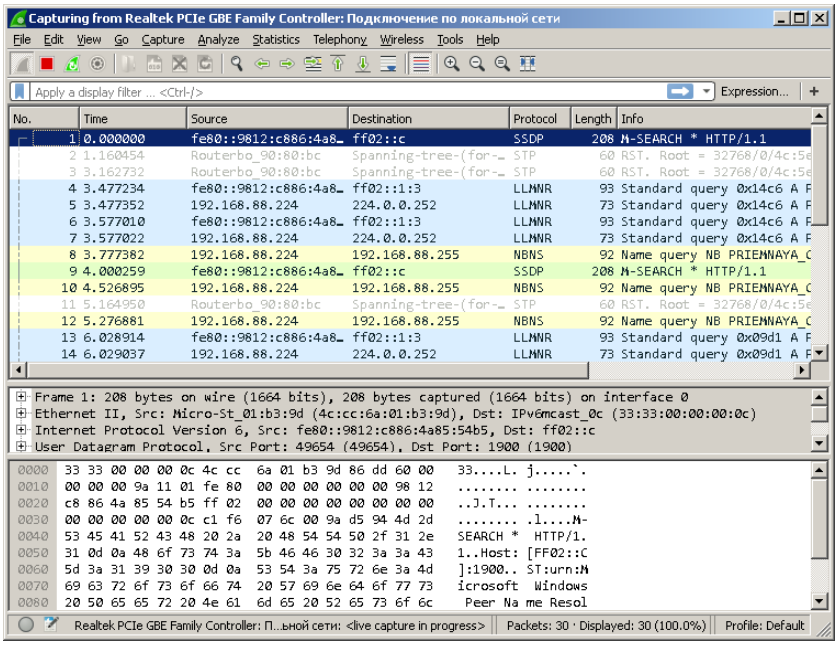


Рис. 3.12. Загальний вигляд інтерфейсу Wireshark під час формування тестових вибірок ОВН

На рис. 3.13 а)–д) показані залежності ІУФР навчання (див. табл. 3.1–3.3) імітаційної моделі ЕС (див. розділ 2) від РКОН ОР – cr. На рис. 3.13 а)–д) середня ділянка відповідає робочій області обраних реалізацій ознак розпізнавання, які мають найбільший показник інформативності, прийнятий за результатами досліджень [80, 109, 132, 137].

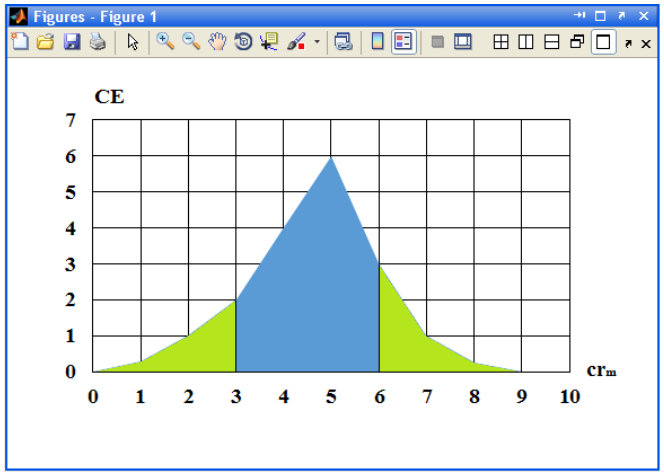
Після формування ББНМ нормальної поведінки системи, відповідно до запропонованого алгоритму (рис. 3.1 табл. 3.1–3.3), будуються бінарні дерева трафіку для мережевих атак та відповідні БВПНМ. Далі визначаються і фіксуються ОВН для системи, що дозволяє сформувати керуючи команди для реагування на відхилення параметрів від розрахункових значень, див. рис. 3.14 а) б).

На рис. 3.15 показані результати отримані в ході імітаційного моделювання алгоритму паралельної кластеризації та оптимізації контрольних відхилень для реалізацій ознак розпізнавання, на прикладі атак класу DoS. Аналогічні результати отримані й для інших класів кібератак.

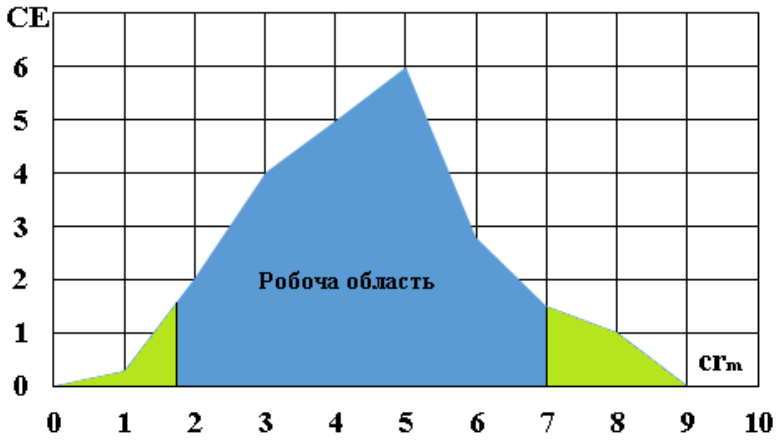
Аналіз результатів імітаційного експерименту з визначення залежності ІУФР навчання ЕС дозволяє зробити наступні висновки:

1) усереднене максимальне значення ІУФР навчання ЕС дорівнює: для атак класу DoS/DDoS $\overline{CE} = 3{,}19$; для атак класу Probe $\overline{CE} = 3{,}15$; для атак класу R2L $\overline{CE} = 2{,}84$; для атак класу U2R $\overline{CE} = 3{,}27$; для вірусних атак (ВА) $\overline{CE} = 2{,}56$;

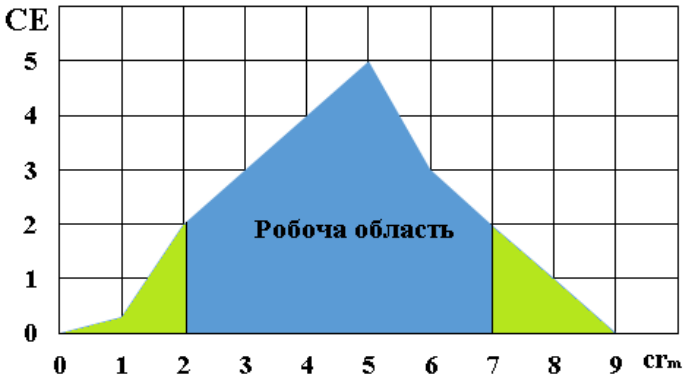
2) усереднене значення оптимального радіусу cr дорівнює у кодових одиницях для класів ОР, представлених у таблиці 3.4 та 3.6, відповідно: для основних класів ОР (гіпотези: hy_{γ_1}; hy_{γ_2}; hy_{γ_3}, таблиця 3.4); для класу $hy_{\gamma 1}$: DoS/DDoS – $cr_1^* = 4$; Probe – $cr_1^* = 3$; R2L – $cr_1^* = 4$; U2R – $cr_1^* = 4$; ВА– $cr_1^* = 5$; для класу $hy_{\gamma 2}$: DoS/DDoS – $cr_2^* = 2$; Probe – $cr_2^* = 1$; R2L – $cr_2^* = 1$; U2R – $cr_2^* = 1$; ВА– $cr_2^* = 2$; для класу $hy_{\gamma 3}$: DoS/DDoS – $cr_3^* = 3$; Probe – $cr_3^* = 3$; R2L – $cr_3^* = 2$; U2R – $cr_3^* = 2$; ВА – $cr_3^* = 3$.



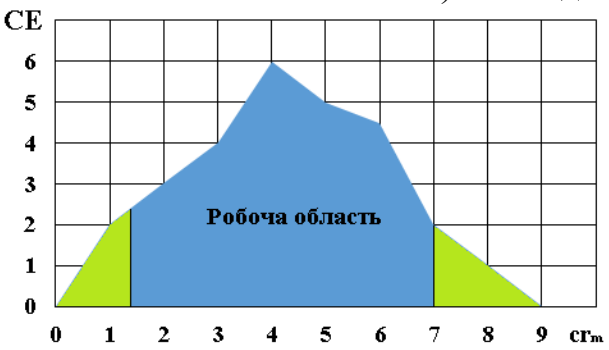
а) ІУФР для атак DoS/DDoS



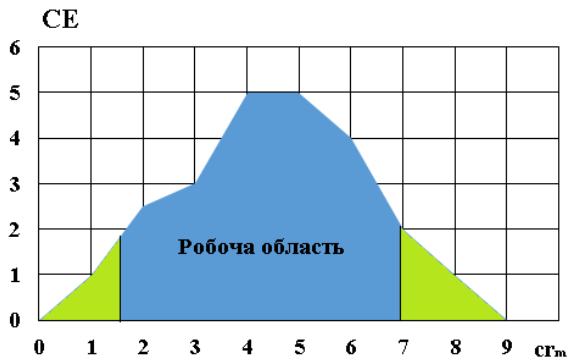
б) ІУФР для атак Probe



в) ІУФР для атак R2L

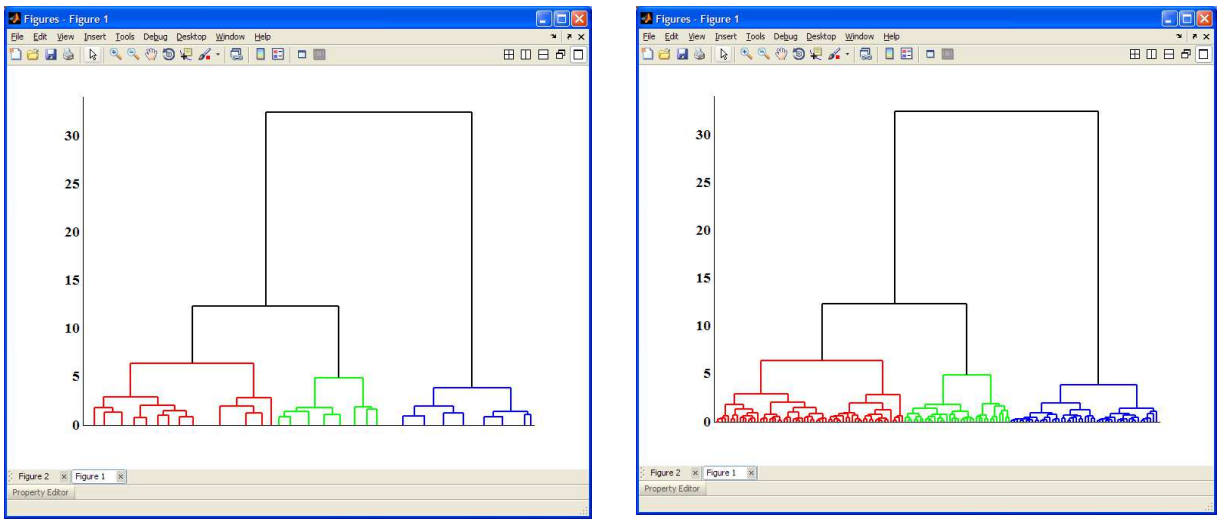


г) ІУФР для атак U2R



д) ІУФР для вірусних атак

Рис. 3.13. Залежності ІУФР навчання імітаційної моделі ЕС від радіусу КОН ОР



а) – нормальний трафік для імітаційної моделі

б) – трафік для випадку розпізнавання мережевої атаки

Рис. 3.14. Структурні характеристики аномального і нормального трафіку

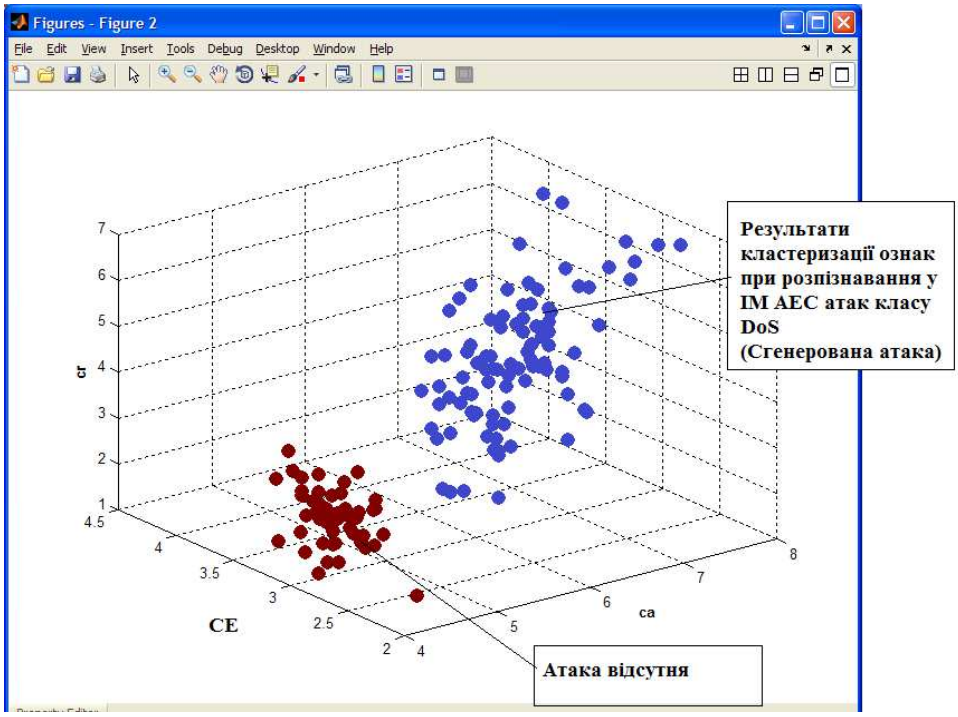


Рис. 3.15. Результати етапів паралельної кластеризації та оптимізації контрольних відхилень для ознак розпізнавання (на прикладі атак DoS)

Значення оптимальних РКОН cr, із урахуванням додаткових гіпотез для розглянутих імітаційних моделей навчання ЕС, наведені у таблиці 3.6.

Таблиця 3.6

Значення оптимального РКОН cr для розглянутих імітаційних моделей навчання ЕС

№	Прийняті гіпотези для ОР	Значення оптимального радіусу КОН cr				
		DoS/DDoS	Probe	R2L	U2R	BA
Основі гіпотези						
1	Основна робоча гіпотеза яка (базова) – $hy_{\gamma 1}$: ознака (ознаки) rc_i ОР (*RS*) та показник IE (див. вираз 2.27) знаходиться у межах звичайного стану КВІС	$cr_1^{опт} = 4-5$	$cr_1^{опт} = 3-4$	$cr_1^{опт} = 4-5$	$cr_1^{опт} = 4-5$	$cr_1^{опт} = 5-6$
2	Гіпотеза $hy_{\gamma 2}$ – ознака (або ознаки) rc_i ОР (*RS*) та показник IE дозволяють зробити висновок, що значення показника IE є меншими на норму	$cr_2^{опт} = 2-3$	$cr_2^{опт} = 1-2$	$cr_2^{опт} = 1-2$	$cr_2^{опт} = 1-2$	$cr_2^{опт} = 2-3$
3	Гіпотеза $hy_{\gamma 3}$ та показник IE дозволяють зробити висновок, що значення показника IE є більшими за норму	$cr_3^{опт} = 3-4$	$cr_3^{опт} = 3-4$	$cr_3^{опт} = 2-3$	$cr_3^{опт} = 2-3$	$cr_3^{опт} = 3-4$
Додаткові гіпотези для імітаційної моделі						
4	Гіпотеза $hy_{\gamma 1}^D$ – вузол u (див. вираз 3.15) демонструє підвищену мережеву активність	$cr_{D1}^{опт} = 4$	$cr_{D1}^{опт} = 4$	$cr_{D1}^{опт} = 3$	$cr_{D1}^{опт} = 3$	–
5	Гіпотеза $hy_{\gamma 2}^D$ – вузол демонструє підвищену мережеву активність	$cr_{D2}^{опт} = 3$	$cr_{D2}^{опт} = 3$	$cr_{D2}^{опт} = 3$	$cr_{D2}^{опт} = 2$	–

Як вже було сказано вище (див. табл. 3.1– 3.3), з метою підвищення ефективності навчання ЕС у складі СІРКЗ та, відповідно, для підвищення показника ІУФР навчання, при здійсненні процедури паралельної оптимізації системи перевірочних/контрольних допустимих відхилень (СКДВ) на ознаки розпізнавання, були отримані наступні результати, див. рис. 3.16.

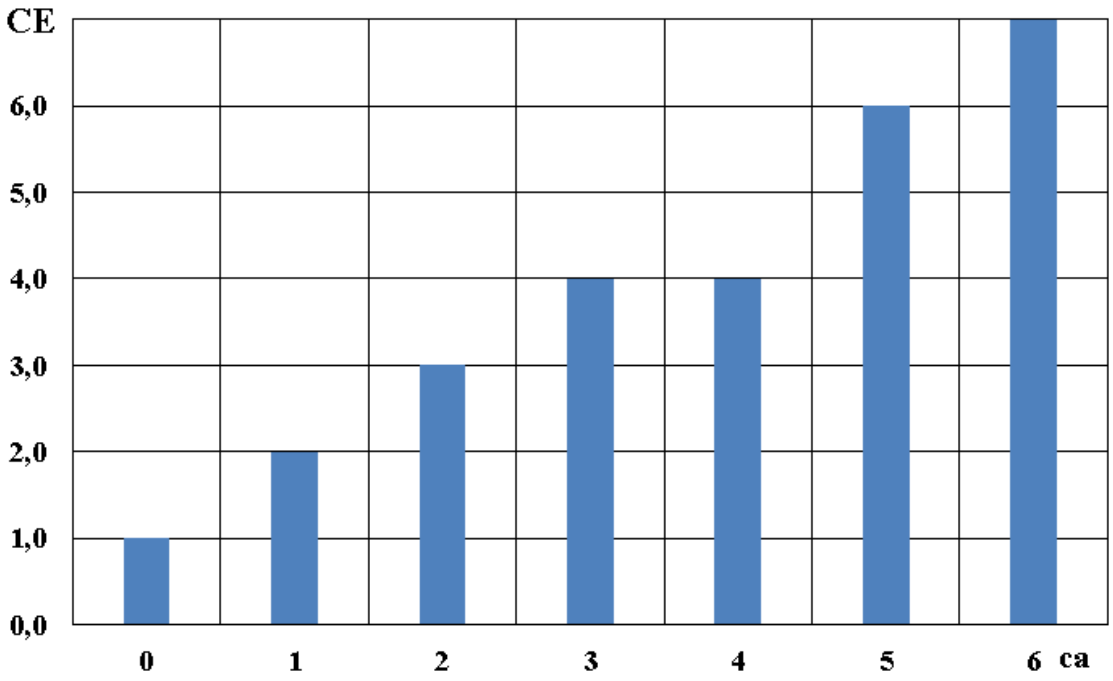


Рис. 3.16. Графік усередненого показника ІУФР навчання від ПДВ для ознак розпізнавання (для мережевих атак)

Як показав аналіз даних отриманих під час імітаційного моделювання ІУФР навчання ЕС, див. рис. 3.13–3.16 та табл. 3.7, квазіоптимальне значення параметра $ca_{n,i}$ СКДВ дорівнює $\text{СКДВ} = 8-16\%$ при максимальному значенні $CE_{\max} = 6{,}16$.

Таким чином, під час імітаційного експерименту підтверджено, що запропоновані моделі та алгоритми кластерізації ознак ОР, які ґрунтуються на ентропійному та інформаційно-дистанційному критерії Кульбака – Лейблера, дозволяють отримувати вхідні нечіткі класифіковані навчальні матриці для ЕС у складі СІРКЗ.

Таблиця 3.7

Результати імітаційного моделювання адаптивної системи інтелектуального розпізнавання із використанням процедури нечіткої кластеризації та паралельної оптимізації контрольних відхилень для ознак кібератак

№	Параметр	Класи об'єктів розпізнавання				
		DoS/ DDoS	Probe	R2L	U2R	BA
1	Кількість реалізацій	65	50	60	50	60
2	Кількість реалізацій ознак (загальна/розглянута)	41/20	41/15	41/15	41/15	20/10
3	Кількість сформованих БНМ	20	15	15	15	10
4	Середня кількість кроків навчання ЕС для формування БНМ (ОВН)	700	900	900	1000	800
5	Середня значення мінімальної кодової відстані між центрами кластерів	$cr_1^* = 4$	$cr_1^* = 3$	$cr_1^* = 4$	$cr_1^* = 4$	$cr_1^* = 5$
6	Оптимальний радіус КОН для гіпотези $hy_{\gamma 1}$ реалізацій ознаки ОР та показник *IE* знаходиться у межах звичайного стану КВІС)	$cr_1^{onm} = 4-5$	$cr_1^{onm} = 3-4$	$cr_1^{onm} = 4-5$	$cr_1^{onm} = 4-5$	$cr_1^{onm} = 5-6$
7	Усереднене максимальне значення ІУФР навчання ЕС (гіпотеза $hy_{\gamma 1}$)	$\overline{CE}_{\max} = 3{,}19$	$\overline{CE}_{\max} = 3{,}15$	$\overline{CE}_{\max} = 2{,}84$	$\overline{CE}_{\max} = 3{,}27$	$\overline{CE}_{\max} = 2{,}56$
8	Ступінь приналежності найкращого, з точки зору ЕС, варіанта Парето–оптимального рішення	6,16	5,90	4,98	5,92	5,02
9	Квазіоптимальне значення контрольних допустимих відхилень для ознак розпізнавання	СКДВ = 8 – 14%	СКДВ = 9 – 15%	СКДВ = 9 – 16%	СКДВ = 8 – 16%	СКДВ = 7 – 14%

3.3. Висновки до третього розділу

Результатом проведених у даному розділі досліджень стали такі висновки:

1) розроблено метод навчання ЕС у складі СІРКЗ, який являє собою ітераційну процедуру пошуку глобального максимуму ІУФР в допустимому діапазоні визначення його функції;

2) запропонована процедура навчання АСР, яка дозволяє попереджати можливі випадки поглинання одним класом ОР базових реалізацій ознак іншого класу, та отримано відповідні предикатні вирази для ЕС здатної до самонавчання;

3) удосконалено метод розбиття простору реалізацій ознак на кластери в ході реалізації процедури розпізнавання кібератак, який відрізняється від існуючих одночасною оптимізацією при обчислення контрольних допусків в ході аналізу ЕС реалізацій ознак ОР, та дозволяє на кожному кроці навчання змінювати ПДВ для всіх реалізацій ознак одночасно;

4) розроблено структурну схему навчання ЕС у складі СІРКЗ для запропонованих етапів паралельної оптимізацією контрольних відхилень для ознак розпізнавання кібератак на КВІС;

5) проведено дослідження за допомогою імітаційного моделювання для моделей та алгоритмів кластерізації реалізацій ознак ОР, які ґрунтуються на ентропійному та інформаційно-дистанційному критерії Кульбака – Лейблера, а також алгоритму оптимізації контрольних допусків на ознаки ОР;

6) встановлено, що за результатами імітаційного експерименту усереднене максимальне значення ІУФР навчання ЕС дорівнює: для атак класу DoS/DDoS $\overline{CE} = 3{,}19$; для атак класу Probe $\overline{CE} = 3{,}15$; для атак класу R2L $\overline{CE} = 2{,}84$; для атак класу U2R $\overline{CE} = 3{,}27$; для вірусних атак (ВА) $\overline{CE} = 2{,}56$;

7) встановлено, що квазіоптимальне значення параметра $ca_{n,i}$ системи контрольних допустимих відхилень дорівнює СКДВ $= 8 - 16\%$ при максимальному значенні ІУФР $CE_{\max} = 6{,}16$;

8) рекомендовано імплементувати запропоновані моделі та алгоритми у розроблювану ЕС «Analyzer of cyberthreats» та провести її тестові дослідження в умовах наближених до реальних режимів роботи діючих систем захисту інформації (СЗІ) для КВІС підприємств.

РОЗДІЛ 4
ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ АДАПТИВНОЇ ЕКСПЕРТНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ КІБЕРАТАК

4.1. Методологія проведення експериментального дослідження адаптивної експертної системи розпізнавання кібератак

Активне впровадження інформаційних технологій в бізнес процеси створює ситуацію в якій багато суб'єктів господарської діяльності змушені забезпечувати постійну доступність своїх інформаційних ресурсів стороннім особам. Тому під час проектування, впровадження та модернізації інформаційно-комунікаційної інфраструктури (ІКС) компаній, підприємств та державних установ, постійно піднімаються питання оцінки прийнятих технічних рішень, які можуть вплинути на продуктивність та стійкість компонентів мережі, засобів кіберзахисту та ІБ, при роботі в умовах штатного і підвищеного навантаження, зокрема впливу кібератак та інших нелегітимних дій з боку зловмисників, направлених проти ІКС або КВІС.

З метою апробації розробленої під час досліджень адаптивної експертної системи розпізнавання кібератак (ЕС «Analyzer of cyberthreats», далі по тексту – ЕС) були проведені тестові дослідження в умовах наближених до реальних режимів роботи діючих систем захисту інформації (СЗІ) для КВІС підприємств.

Тестові дослідження розробленої ЕС, за погодженням з керівництвом, були проведені на підприємствах та закладах – Товариство з обмеженою відповідальністю «Інформаційна безпека», та Чернігівський національний технологічний університет (далі по тексту, відповідно – Замовник 1 та Замовник 2), відповідні акти впровадження наведені в додатку А.

Згідно з методикою проведення тестових досліджень початковими даними були:

1. Перелік IP-адрес хостів, що утворюють зовнішній периметр ІС (КВІС).

2. Перелік адрес електронної пошти співробітників, (адреси були отримані у результаті репрезентативної вибірки серед співробітників з різних структурних підрозділів Замовника;

3. Дані про встановлені ІС, ПЗ, СКБД та АСК підприємств.

Була використана методика, що дозволила найбільш повно змоделювати дії потенційного порушника [71, 108, 109, 112], зокрема:

пасивний збір відомостей про ІС Замовника з відкритих джерел;

активний збір відомостей про ІС Замовника (підключення до хостів зовнішнього периметра);

перевірка можливості проникнення в ІС Замовника за допомогою використання уразливостей мережевих служб, запущених на хостах зовнішнього периметра;

перевірка можливості проникнення в ЛОМ та зменшення продуктивності ІС Замовника за допомогою троянської програми.

У якості потенційних порушників політики ІБ (атакуючої сторони), розглядалася група осіб, які перебувають у змові і у результаті навмисних дій можуть реалізувати різноманітні кіберзагрози для ІС, спрямовані на інформаційні ресурси і завдати моральної та/або матеріальної шкоди інтересам Замовника.

Структура формальної моделі процесу організації та проведення тестових експериментальних досліджень ЕС «Analyzer of cyberthreats», яка працює у складі ЛОМ і ІС (КВІС) підприємств представимо у вигляді наступного графа, рис. 4.1. Смислове значення вершин графа подано у таблиці 4.1.

Вершини графа відповідають станам ІС. Тобто, зміна станів настає відповідно до дій експериментатора, який виконує роль зловмисника і намагається в ході атаки подолати захист ІС. Дуги графа відповідають зв'язкам між етапами подолання рубежів кіберзахисту ІС, і порядок

виконання певних дій з боку порушника (порушників): $G = (S, P)$, де $S = (S_1,..., S_N)$; $P = (P_1,..., P_M)$.

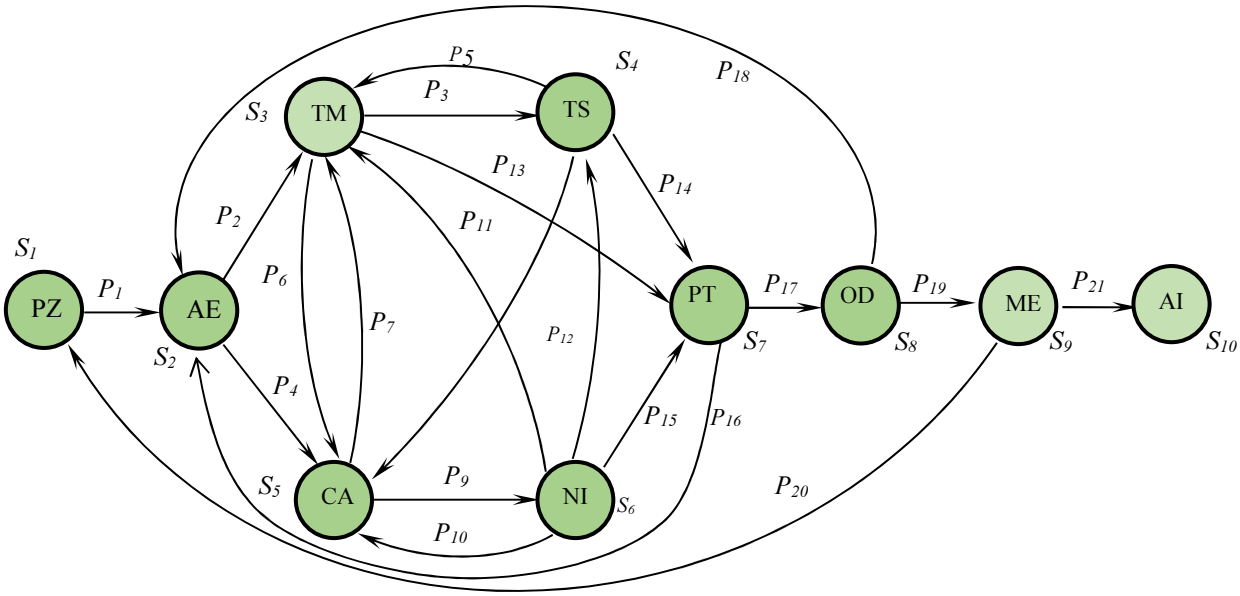


Рис. 4.1. Структура формальної моделі процесу організації та проведення експерименту

Відповідно до постановки завдання (PZ), складено алгоритм (AE) експериментальних досліджень ЕС у ході розпізнавання нелегітимних дій зловмисників, які цілеспрямовано намагаються подолати захист ІС. Одержаний алгоритм дозволяє намітити дії, необхідні для здійснення поставленого завдання з подолання рубежів захисту ІС. Також на підставі розробленого алгоритму, проводиться вибір засобів вимірювання мережевого трафіку (ТМ) в ЛОМ під час штатних режимів роботи ІС та у ході здійснення типових кібератак, які належать класам – Probe, U2R, R2L, Dos/DDos. На наступному етапі виконується складання алгоритму управління (СА) технічними засобами вимірювання мережевого трафіку в ході тестів на проникнення та апробації ЕС.

Таблиця 4.1

Смислове значення вершин графа G

№	Позначення	Смислове значення
1	S_1	постановка завдання експерименту, в ході якого реалізується тест на проникнення (PZ)
2	S_2	складання алгоритму експериментів для тесту на проникнення в ІС (AE)
3	S_3	вибір засобів вимірювання трафіку (TM)
4	S_4	вибір засобів захисту інформації (СЗІ) (TS)
5	S_5	побудова алгоритму управління ЗЗІ (CA)
6	S_6	налагодження засобів вимірювання (NI)
7	S_7	проведення експерименту (тест на проникнення) (PT)
8	S_8	опрацювання отриманих даних (OD)
9	S_9	моделювання елементів ІС на основі експериментальних даних (OD)
10	S_{10}	аналіз зміни продуктивності ІС в ході тестів на проникнення (AI)

В ході тестових досліджень ЕС «Analyzer of cyberthreats», для скорочення розміру даних при навчанні адаптивної системи, й аналізу мережевого трафіку у ІС, використовувалися моделі, представлені в розділах 2 і 3 дисертаційної роботи. Основним критерієм відбору параметрів є рівень статистичної значущості [8]. Таким чином, після проведення процедури мінімізації загальної кількості параметрів для аналізу та відбору найбільш інформативних характеристик, число параметрів скоротилося з 41 (таблиця 4.2) до 5–7. Як було встановлено в роботах [7, 80, 142], більш інформативні чинники містять 99% даних, необхідних для розпізнавання атаки.

У процесі розробки алгоритму управління і настроювання засобів вимірювання показників продуктивності ІС (NI), можна коригувати склад і функцій окремих технічних засобів (P_7, P_{12}). У результаті отримаємо експериментальні дані, а також, ще на першому етапі, виявимо можливі помилки в загальному алгоритмі проведення тестів ЕС. Для усунення помилок виконується корекція (P_{16}). Отримані експериментальні результати

піддаються опрацюванню (OD), після якого отримуємо масиви даних для проведення аналізу й подальшого моделювання роботи мережі та інформаційної системи.

Таблиця 4.2

Параметри мережевого трафіка для побудови ОВН

№ з/п	Параметр	Опис
1.	*duration*	Тривалість (у секундах) з'єднання
2.	*protocol_type*	Тип протоколу (TCP, UDP, etc.)
3.	*service*	Атакований сервіс
4.	*src_bytes*	Кількість байтів від джерела до призначення
5.	*dst_bytes*	Кількість байтів відповіді клієнту
6.	*flag*	Прапорці з'єднання
7.	*land*	1, якщо з'єднання від/до того самого хоста/порта
8.	*wrong_fragment*	Кількість „хибних" фрагментів
9.	*urgent*	Кількість термінових пакетів
10.	*hot*	Кількість „гарячих" індикаторів
11.	*num_failed_logins*	Кількість невдалих спроб реєстрації
12.	*logged_in*	1, якщо успішний вхід в систему; 0 неуспішне
13.	*num_compromised*	Кількість „компроментуючих" умов
14.	*root_shell*	1, якщо root shell отриманий; інакше 0
15.	*su_attempted*	1, якщо виконувалась „su root" ; інакше 0
16.	*num_root*	Кількість „root" доступів
17.	*num_file_creations*	Кількість операцій створення файлів
18.	*num_shells*	Кількість запитів на надання оболонки
19.	*num_access_files*	Кількість операцій на доступ до контролю файлів
20.	*num_outbound_cmds*	Кількість вихідних команд для FTP сесії
21.	*is_hot_login*	1, якщо логін належав до „гарячого" списку
22.	*is_guest_login*	1, якщо „гостьовий" вхід

продовження табл. 4.2

23.	*count*	Кількість з'єднань на хост в поточній сесії за останні 2 с
24.	*serror_rate*	% з'єднань що мали „SYN” помилки
25.	*rerror_rate*	% з'єднань що мали „REJ” помилки
26.	*same_srv_rate*	% з'єднань що мали однаковий сервіс
27.	*diff_srv_rate*	% з'єднань на різні сервіси
28.	*srv_count*	Кількість з'єднань на такий самий сервіс за останні 2 с
29.	*srv_serror_rate*	% з'єднання з помилкою в „SYN” пакеті
30.	*srv_rerror_rate*	% з'єднання, що мають „REJ” помилки
31.	*srv_diff_host_rate*	% з'єднання від інших хостів
32.	*dst_host_count*	Кількість з`єднань до локального хоста, встановлених віддаленою стороною
33.	*dst_host_srv_count*	Кількість з`єднань до локального хоста, встановлених віддаленою стороною та використовуючих одну службу
34.	*dst_host_same_srv_rate*	% з`єднань до локального хоста, встановлених віддаленою стороною та використовуючих одну службу
35.	*dst_host_diff_srv_rate*	% з`єднань до локального хоста, встановлених віддаленою стороною та використовуючих різні служби
36.	*dst_host_same_src_port_rate*	% з`єднань до даного хоста при поточному номері порту джерела
37.	*dst_host_srv_diff_host_rate*	% з`єднань до служби різних хостів
38.	*dst_host_serror_rate*	% з`єднань з помилкою типу SYN для даного хост-приймача
39.	*dst_host_srv_serror_rate*	% з`єднань з помилкою типу SYN для даної служби приймача
40.	*dst_host_rerror_rate*	% з`єднань з помилкою типу REJ для даного хост-приймача
41.	*dst_host_srv_rerror_rate*	% з`єднань з помилкою типу REJ для даної служби приймача

На цьому етапі проведення досліджень можуть виникати похибки в складанні алгоритму тесту і, відповідно, необхідність його корекції (P_{18}). Наступним етапом досліджень є моделювання процесів передачі запитів і

отримання даних (МЕ) у ІС. У разі незадовільного результату роботи моделі можливе коригування поставленого для ЕС завдання із розпізнавання (P_{20}).

Результат дослідження захищеності та продуктивності ІС (АІ) завершує процес експериментальних досліджень тесту на проникнення.

Тестування роботи ЕС яка була розгорнута на ЕОМ у складі ІС підприємства Замовника, проводилось за наступних умов, табл. 4.3.

Таблиця 4.3

Параметри які досліджувалися при тестуванні ЕС

Параметр	Опис
service_no	Кількість ЕОМ (ПК) у складі досліджуваних сегментів ЛОМ та ІС підприємств Замовників
service k	Кількість одночасно працюючих ЕОМ у складі ЛОМ
service_sk	Передбачувана кількість запитів, які очікують обслуговування на серверах ІС
service_ qs	Кількість спроб нелегітимного доступу до ІС та ЛОМ (з урахуванням генерованих нападником (*i*))
*service_* um	Кількість ЕОМ задіяних у процесах обробки критично важливої інформації
Intensity_flow_ requests	Інтенсивність потоку запитів, що надходять на сервери ІС
nominal_capacity	Номінальна пропускна здатність середовища передачі даних
interframe_ interval	Міжкадровий інтервал
time to respond	Час очікування відповіді під час виконування операцій у ІС

Методика проведення експериментів під час тестів включала:

елементи вимірювального експерименту продуктивності ІС у штатних режимах роботи та під час кібератаки;

тести для окремих модулів розробленої ЕС;

отримання характеристик структурованої кабельної системи обчислювальної мережі ІС підприємств Замовників;

первинне дослідження мережі (одержання списку імен ЕОМ, доступних портів та ін.);

виявлення недоліків спеціального ПЗ та архітектури мережі підприємств Замовників;

підключення аналізатора протоколів до досліджуваного обчислювального комплексу ЛОМ, ІС та розробленої ЕС;

інтерпретація результатів тестових досліджень на вимірювань.

У ході тестових досліджень ЕС також розглядалося завдання розпізнавання нелегітимних спроб отримання та управління правами доступу до інформаційних ресурсів Замовників. Змістовна постановка задачі має такий вигляд:

Вектор $V_i = \{v_i\}$ реалізацій ознак об'єктів доступу (ОД) до ІС (КВІС):

$$v_i = \begin{cases} 1, \textit{ якщо до вузла } v_i \textit{ дозволено загальний доступ}; \\ 0, \textit{ у протилежному випадку}. \end{cases} \quad (4.1)$$

Матриця $M_{СД} = [m_{СД_{j,k}}]$ розподілу суб'єктів доступу (СД) до локальних вузлів (ЛВ) ІС (КВІС):

$$m_{СД_{j,k}} = \begin{cases} 1, \textit{ якщо } sd_j \in ut_k; \\ 0, \textit{ у протилеж ному випад ку}. \end{cases} \quad (4.2)$$

Матриця $M_{ОД} = [m_{ОД_{j,k}}]$ розподілу ОД до локальних вузлів (ЛВ) ІС (АС):

$$m_{ОД_{j,k}} = \begin{cases} 1, \textit{ якщо } rd_j \in ut_k; \\ 0, \textit{ у протилежному випадку}. \end{cases} \quad (4.3)$$

Критерії кібербезпеки ІС (КВІС $Z_0 = \{z_{0_{i,j}}\}$:

$$z_{0_{i,j}} = \begin{cases} 1, \textit{ якщо } rd_{\mathrm{i}} \textit{ розміщено на вузлі } \mathrm{um_k}; \\ 0, \textit{ у протилежному випадку}. \end{cases} \quad (4.4)$$

де $RD = \{\mathrm{rd_i}\}; \mathrm{i} = \overline{1,\mathrm{I}}$ – множина об'єктів доступу;

$SD = \{\mathrm{sd_j}\}; \mathrm{j} = \overline{1,\mathrm{J}}$ – множина суб'єктів доступу у ІС (КВІС);

$UM = \{\mathrm{um_k}\}; k = \overline{1,\mathrm{K}}$ – множина локальних вузлів.

Зведена характеристика ймовірних порушників наведена в таблиці 4.4.

Таблиця 4.4

Характеристики ймовірних порушників політики ІБ

Класифікація	Характеристика
За мотивом порушення	Порушення цілісності, конфіденційності, доступності з корисливою чи іншою метою.
За рівнем інформованості та кваліфікації порушника	Порушник має: 1) високий рівень знань; 2) достатні знання для збору інформації, застосування відомих експлойтів та написання власного ПЗ для здійснення кібератак; 3) порушник (и) не є авторизованим користувачем ІС (КВІС).
За місцем дії	Без безпосереднього (фізичного) доступу на територію об'єкта (зовнішній порушник). Порушник діє віддалено, через мережу Інтернет

В ході досліджень ЕС були зроблені наступні припущення:

1) обчислювальна мережа підприємства Замовника буде здійснювати передачу запитів Ethernet в умовах конкурентного доступу модулів до поділюваного середовища передачі даних, при цьому вплив колізій на роботу обчислювальної мережі не виключається;

2) передбачуване інформаційне навантаження – *Intensity_flow_ requests =340–350* запитів/с.

На підготовчому етапі робіт з виконання тесту на проникнення було складено опис сервісів, що надаються ІС співробітникам підприємства і користувачам глобальної мережі Інтернет, наприклад, перегляд замовлень, попереднє оформлення договорів та ін. Така попередня оцінка дозволяє виділити основні напрямки, що в першу чергу підлягають аналізу з точки зору забезпечення ІБ.

Було зроблено припущення, що типові сервіси даних ІС – це сайт компанії, електронна пошта, системи доступу для віддалених співробітників та клієнтів.

Дослідження продуктивності ЛОМ і ІС підприємств виконувалося за допомогою підключення робочої станції аналізатора протоколів Network Instruments Observer 9.1 до мережі підприємства. Також використовувалися програми TRACEROUTE і TRACEPATH.

Для виявлення в АСК пристроїв, що взаємодіють по протоколу Modbus використовувалася утиліта PLCScan.

При проведенні тестових досліджень ЕС «Analyzer of cyberthreats», яка дозволяє оцінювати стан ІБ ІС (КВІС), окремо ставилось питання оцінювання ступеню захищеності систем від потенційно небезпечних кібератак, а також про розробку рекомендацій, виконання яких може сприяти підвищенню рівня ІБ. Відповідно до цих завдань у склад ЕС імплементовані модулі які дозволяють:

автоматизувати процедуру проведення аудиту ІБ ІС (КВІС);

покращити процедуру розпізнавання загроз ІБ;

отримувати експертну інформацію про стан комп'ютерів у мережі підприємства;

сканувати запущені програми на ЕОМ підприємства;

визначати рівнів ІБ окремих ЕОМ (ПК або АРМ) у складі ІС (КВІС);

полегшити роботу експертів з ІБ;

використовувати накопичений раніше досвід з оцінювання стану ІБ;

автоматизувати процес ухвалення рішення по вибору оптимальної стратегії управління кіберзахистом КВІС (КВКС);

оцінювати поточні ризики НСД до ІС підприємства;

представити рекомендації із підвищення рівня захищеності ІС;

зменшити час на проведення перевірок та аудиту стану ІБ для ІС (КВІС).

Для представлення знань в ЕС використовується фреймова модель, для прийняття рішення – прямий логічний висновок.

В основу ЕС покладено припущення про те, що елементи множини функцій безпеки можуть не повністю забезпечувати виконання вимог ІБ на підприємстві, а отже, призводити до зростання показника поточних інформаційного ризиків [121, 139]. Задається рівень поточного ризику, який вважається прийнятним і не вимагає застосування дорогих заходів для протидії спробам НСД у КВІС.

Програма включає в себе кілька модулів, що можуть функціонувати і як єдиний комплекс, і у вигляді самостійних програмних продуктів, зокрема, розглянутих у розділах 2 та 3.

Структурна схема основних модулів ЕС «Analyzer of cyberthreats», яка я частиною АСР, представлена на рис. 4.2. На рис. 4.3 показано приклад алгоритму роботи модулю з аналізу та додавання загроз у базу знань ЕС для оцінювання ІБ ІС або КВІС.

Для розробки інтерфейсів користувача та експерта, а також модулів розпізнавання кібератак, обчислення ступеня захищеності та ін. функціональних модулів використовувалася мова та середовище програмування Delphi. Для проектування ЕС обрана програма-оболонка CLIPS.

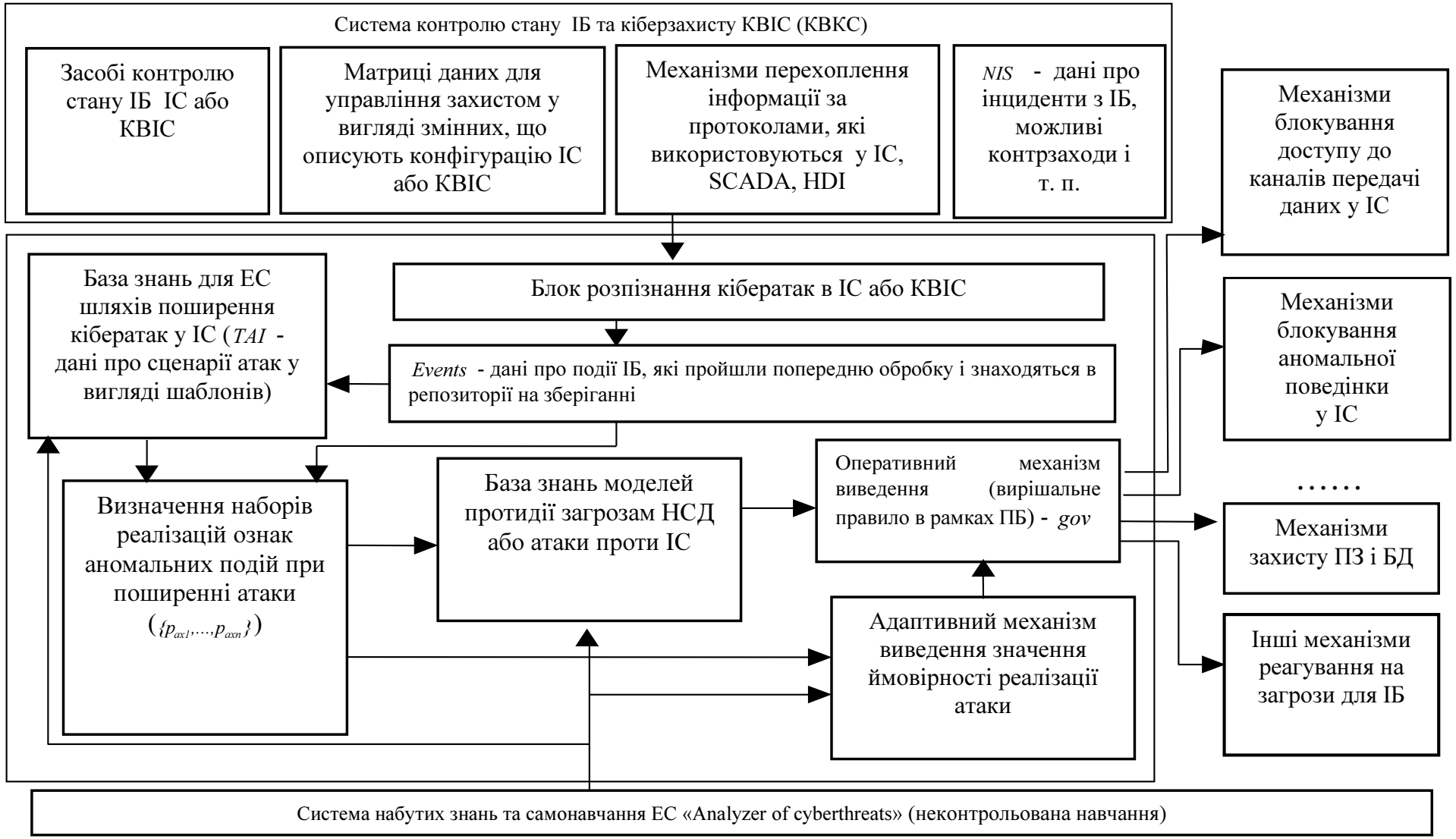


Рис. 4.2. Структура схема ЕС у складі АСР кіберзагроз (неконтрольоване навчання)

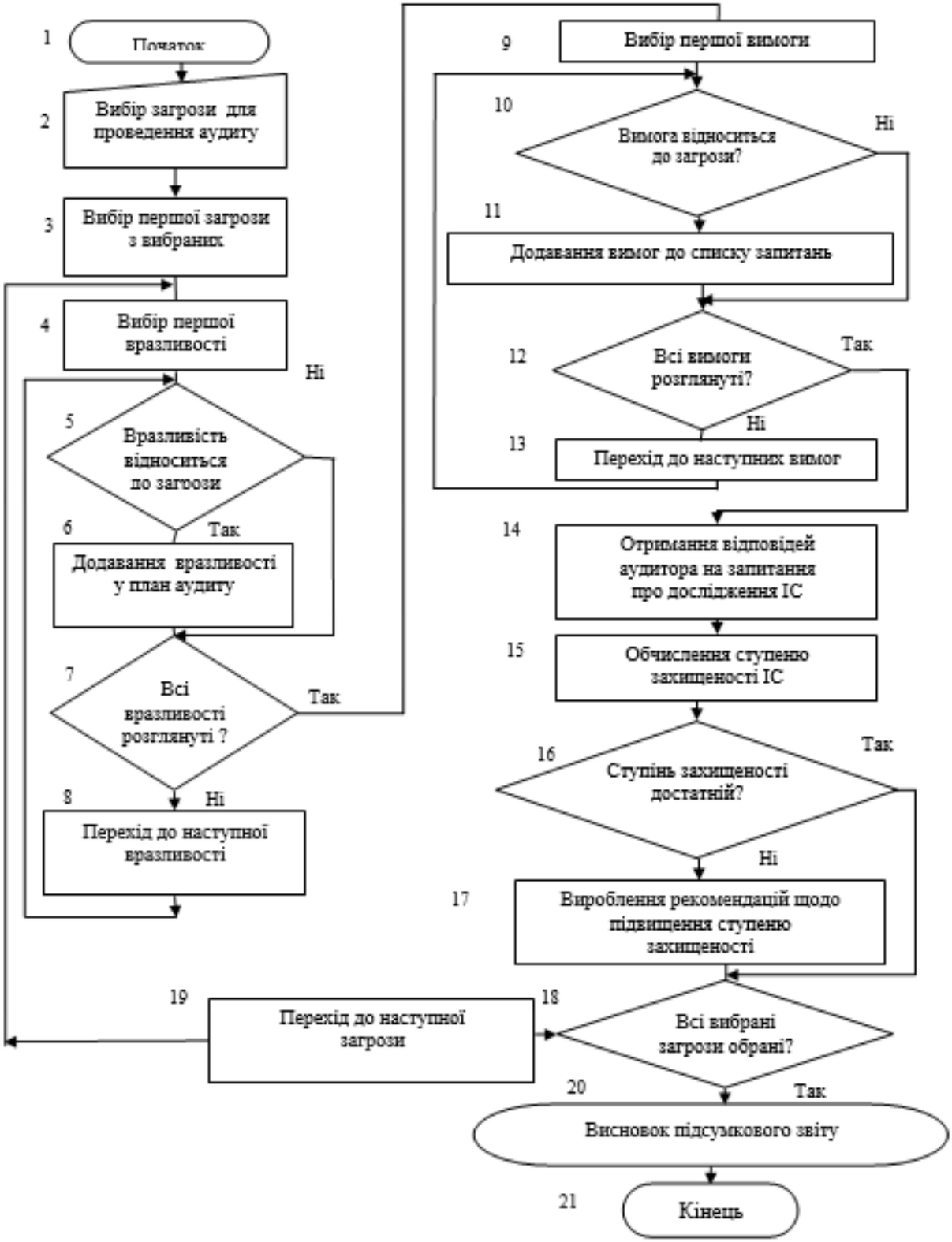


Рис. 4.3. Алгоритм аналізу та додавання загроз у базу знань ЕС

Схема взаємодії модулів програми показана на рис. 4.4.

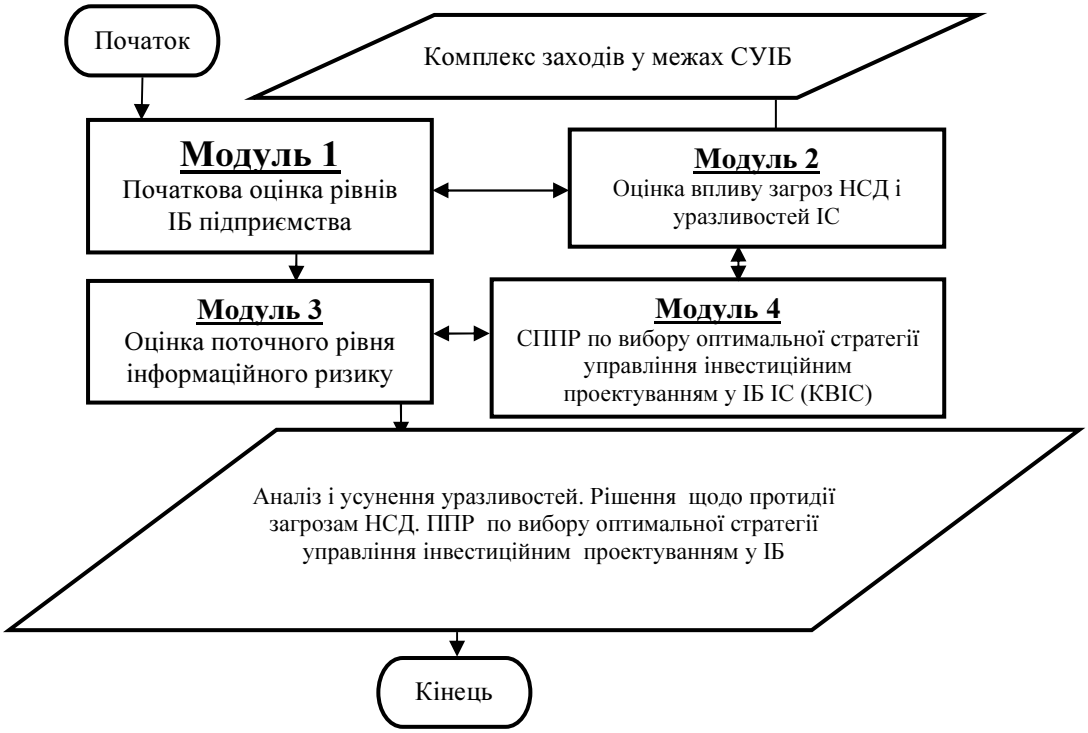


Рис. 4.4. Схема взаємодії модулів ЕС «Analyzer of cyberthreats»

На рис. 4.5–4.9 показані основні форми та закладки ЕС Аналізатор загроз. Основні коди ЕС «Analyzer of cyberthreats» наведені в додатку Б.

Інтерфейс користувача ЕС призначений для фахівців з ІБ. Через інтерфейс аналітик стану ІБ ІС або аудитор отримує необхідну інформацію та передає у ЕС запитувані дані: відомості про вимоги ІБ, задіяні системи захисту інформації та ін. Через цей же інтерфейс здійснюється первинний вибір та аналіз загроз ІБ за ознаками. Вся інформація, що вводиться користувачем через інтерфейс, передається в робочу пам'ять. ЕС використовує інтерфейс користувача для надання підсумкових звітів з результатами аналізу стану ІБ та виробленими рекомендаціями.

Інтерфейс експерта призначений для передачі знань експертів з ІБ у БЗ, а також, для коригування знань та правил розпізнавання кібератак. Через інтерфейс здійснюється й зміна модулів прийняття рішення. Це відбувається тільки в тому випадку, якщо в роботі ЕС виявлені помилки.

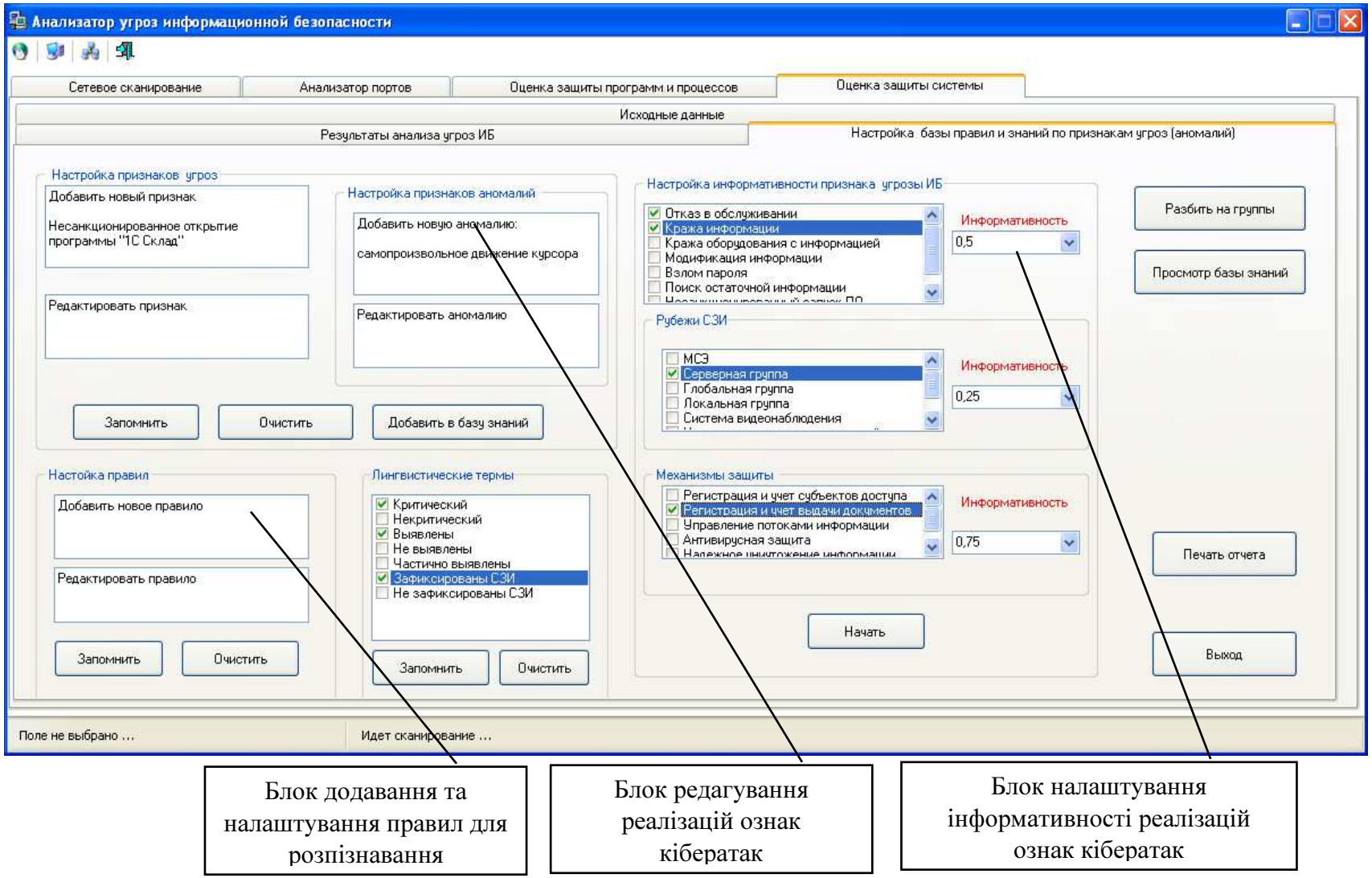
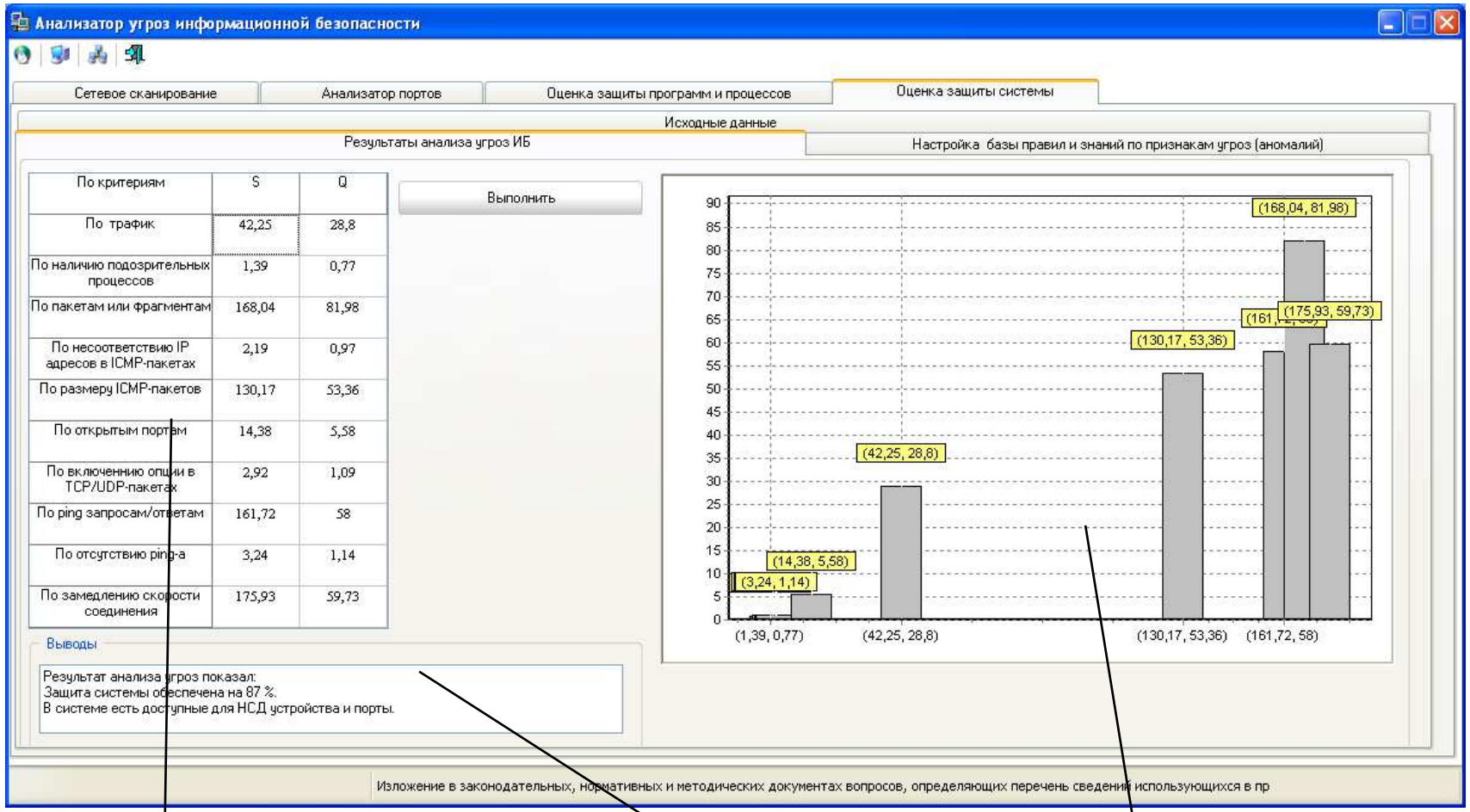


Рис. 4.5. Закладка для налаштування правил розпізнавання та оцінювання кібератак



Блок оцінювання параметрів безпеки ІС для мережевих з'єднань

Блок висновку оцінювання стану ІБ для ІС

Графічне представлення тестових параметрів ІБ для ІС

Рис. 4.6. Закладка для представлення результатів аналізу виявлених кібератак

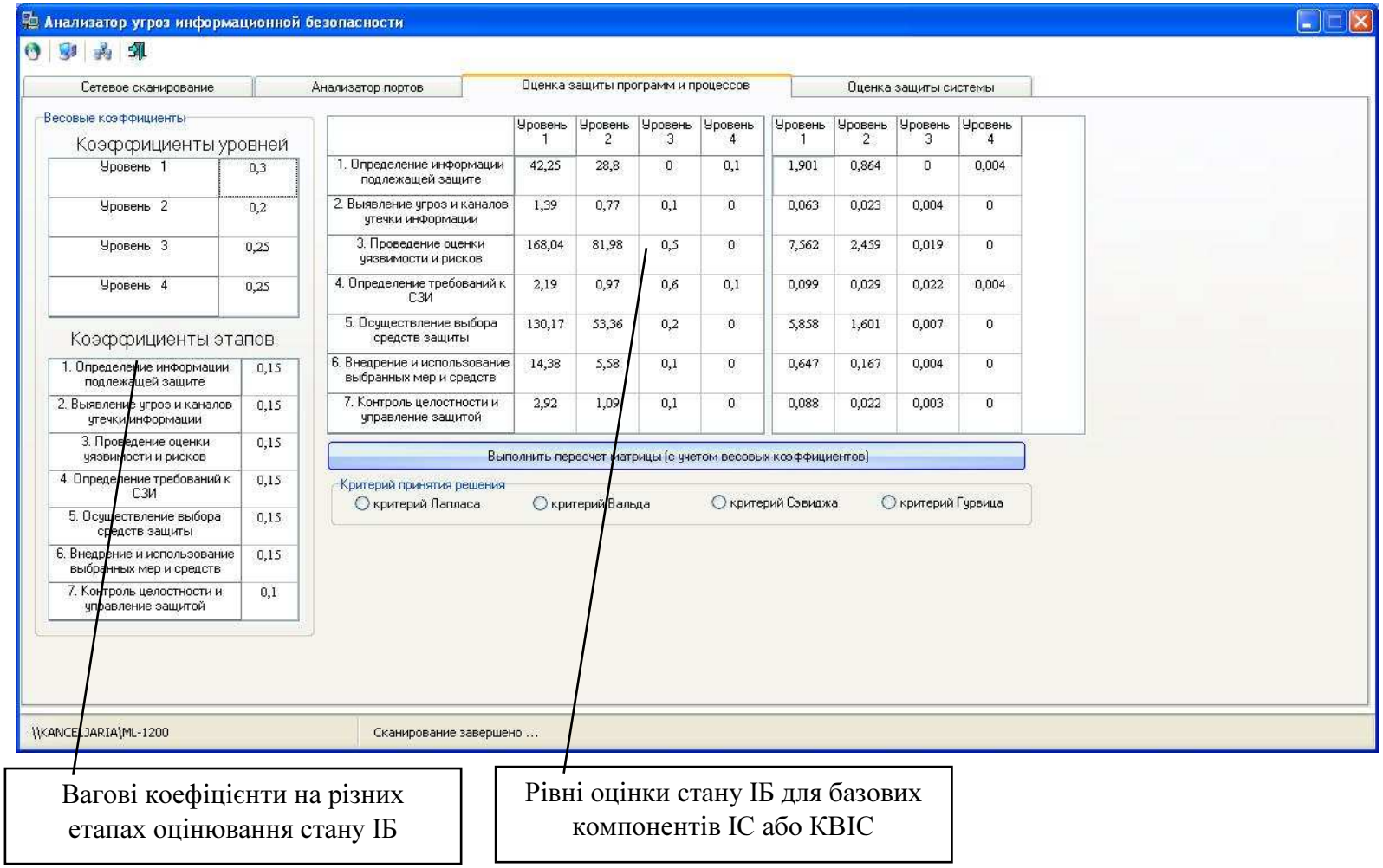


Рис. 4.7. Закладка для представлення результатів оцінювання стану ІБ для базових компонентів ІС або КВІС

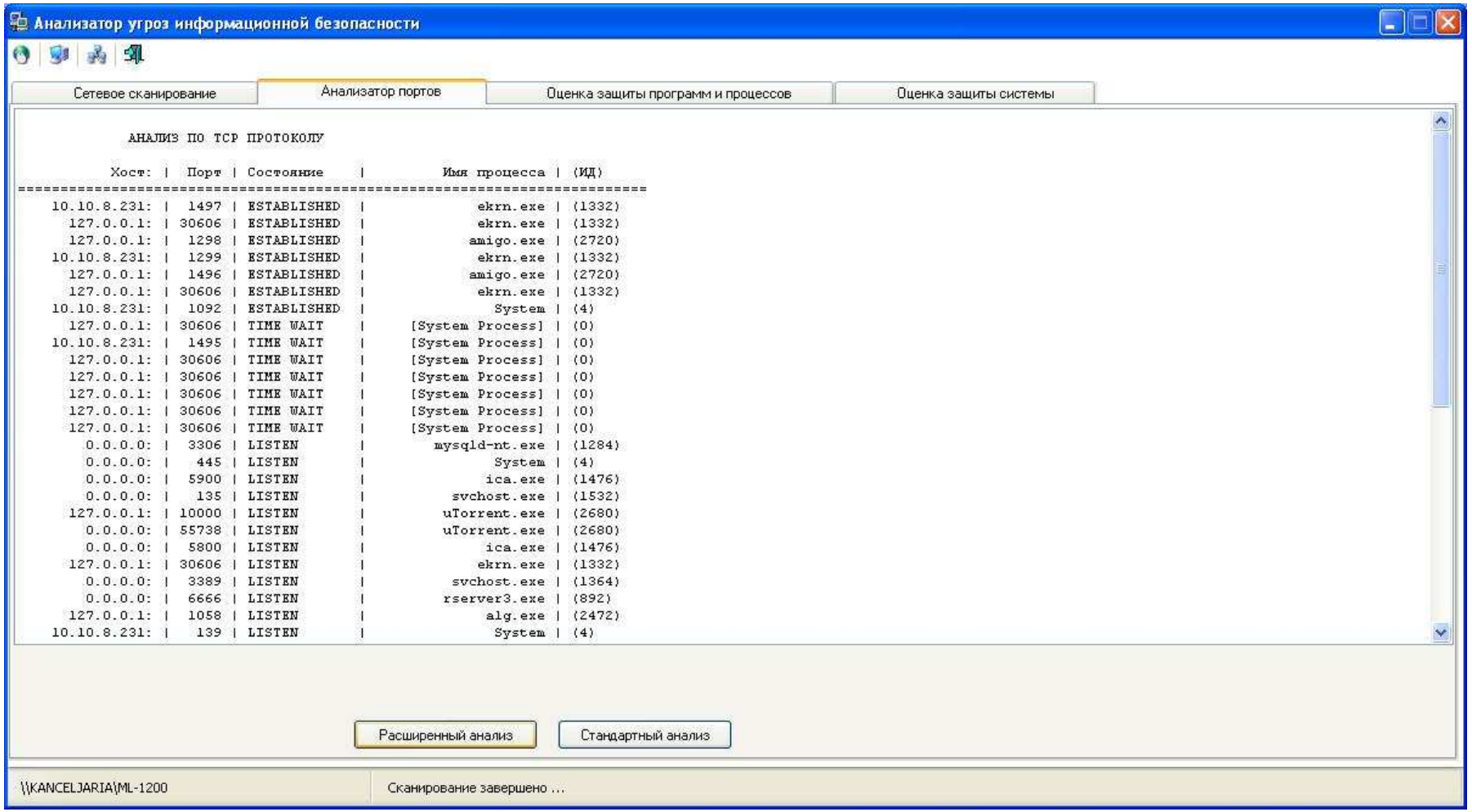


Рис. 4.8. Закладка для представлення результатів оцінювання портів у ЕОМ у складі ІС для виявлення несанкціонованих мережевих з’єднань

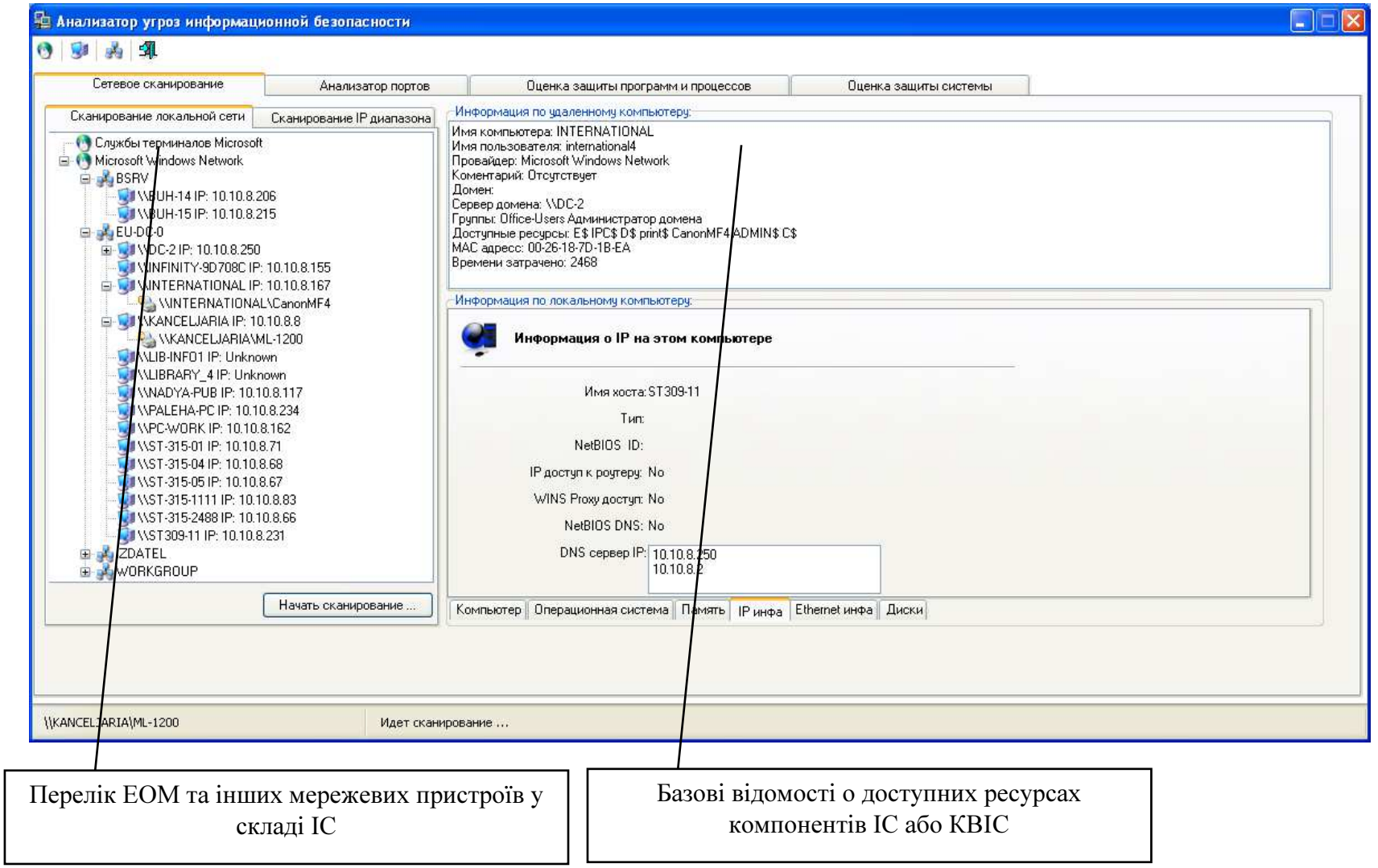


Рис. 4.9. Закладка для представлення результатів сканування окремих ЕОМ у складі ІС

Багаторівнева модель оцінювання стану ІБ на першому етапі роботи ЕС, відповідає мінімальній активації потенційних механізмів захисту та повноті інформаційного поля відомих загроз для ІС.

На другому етапі роботи ЕС, БЗ поповнюється новою інформацією, шляхом переведення механізмів захисту ІС із статусу «потенційні загрози» у статус «активні загрози» і прив'язки активованого механізму до відповідного ешелону СЗІ. Число елементів в підмножині заданих кібератак, збільшується, як за рахунок включення елементів з множини відомих об’єктів, так і за рахунок поповнення множини раніше невідомих загроз [7, 32, 95, 132, 148, 194, 210]. Далі можливе розширення множини потенційних механізмів захисту ІС за рахунок подальшого опису у вигляді продукційних правил і подальшої реалізації раніше відсутніх механізмів захисту ІС. При подальшій адаптації ЕС до потреб реальної ІС (КВІС) відбувається її навчання, спрямоване на безпомилкове розпізнавання кібератак за мінімальною кількістю реалізацій ознак. Аналіз додаткового продукційного правила дозволяє сформувати специфікацію на використання відсутнього в ІС засобу або механізму захисту інформації.

Як цільова функція в ЕС була прийнята величина очікуваного збитку від НСД або кібератаки, що визначається через міру розбіжності між реальним та оптимальним механізмами розмежування доступу в ІС:

$$W = \sum_{i=1}^{I} \sum_{j=1}^{J} p_{i,j} \cdot \Delta z_{0_{i,j}}, \tag{4.5}$$

де $P_{i,j} = \{p_{i,j}\}, \mathrm{i} = \overline{1,\mathrm{I}}, \mathrm{j} = \overline{1,\mathrm{J}}$ – матриця шкоди, зумовленої можливістю НСД до ресурсів, при чому, елементи $\{p_{i,j}\}$ визначаються ступенем конфіденційності інформації в ІС і профілем користувача;

$\Delta z_{0_{i,j}}$ – критерій кібербезпеки КВІС.

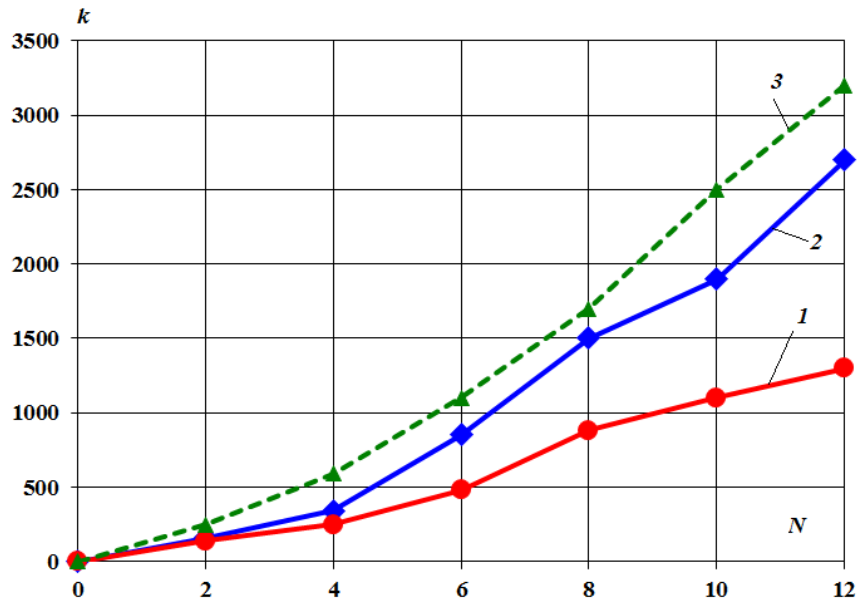
4.2. Результати експериментальних досліджень

Під час проведення тестових досліджень ЕС у складі АСР, в якості вхідних даних для навчання та тестування використовувалася база даних KDD Cup Data [140, 144, 162].

Для перевірки ефективності запропонованої моделі, проведена серія експериментів для основних атак, показаних в табл. 4.5.

На рис. 4.10 – 4.14 показані результати тестування ЕС «Analyzer of cyberthreats» для різник класів атак, спрямованих на компоненти ІС (КВІС).

На рис. 4.10 показані результати тестування запропонованої моделі для розпізнавання атак типу «Відмова в обслуговуванні» у порівнянні із найбільш поширеними у комерційних СВА та ЕС моделях, які базуються на прогнозуванні станів ІС або КВІС та моделі яка передбачає послідовний перебір реалізацій ознак кібератак [80, 137, 162, 210].



(***N*** – кількість ознак; ***k*** – кількість кроків навчання АСР)

1 – АСР; 2 – методи прогнозування станів; 3 – послідовний перебір ознак

Рис. 4.10. Порівняльна ефективність запропонованої моделі для розпізнавання атак типу «Відмова в обслуговуванні»

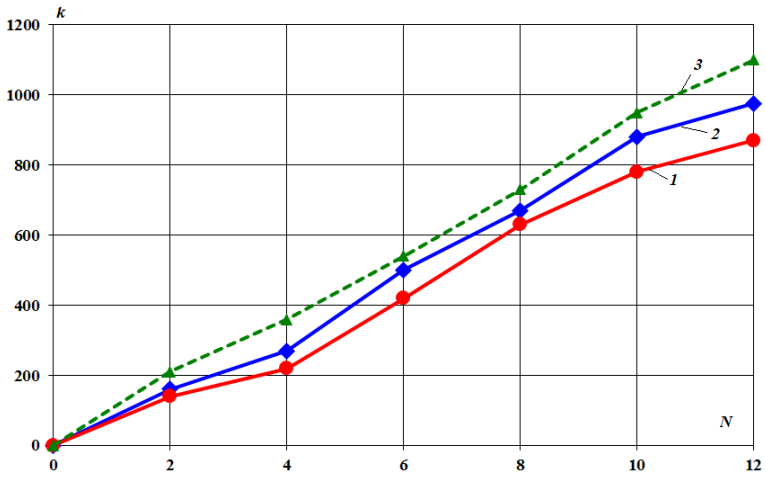
Таблиця 4.5

Середня кількість правил, матриць і кроків навчання АСР для розпізнавання типових класів кібератак в КВІС (КВКЗ)

Клас об'єктів для розпізнавання (Кібератак) *	Кількість реалізацій ознак **	Середня кількість правил, матриць та кроків для навчання (Правила / Матриці / Кроки для навчання)		
		Моделі та алгоритми послідовного перебору ознак ***	Статистичні моделі прогнозування станів ****	Модель, заснована на МІЕТ та ЕК
Мережеві атаки через корпоративну систему	11	200/30/2000	350/65/2000	60/10/2000
Атаки на стандартні компоненти ПЗ КВІС (КВКС)	19	350/50/3500	450/35/3500	30/15/1500
Мережева розвідка	15	320/40/2500	120/30/2500	70/20/2000
Атаки, спрямовані на підбір паролів	12	230/15/1500	180/25/1300	25/20/1500
Атаки типу Man-in-the-Middle	9	300/40/4000	350/30/3000	40/20/2000
DoS/DDoS – атаки	9	150/25/2500	170/25/2000	30/15/1500
Вірусні атаки	21	400/50/2700	400/60/2500	35/25/1700
Атаки на ERP системи через протокол HARD	5	170/30/2700	210/50/2300	60/35/1900
Атаки на компоненти ЛОМ	9	260/25/2400	200/40/2500	45/35/2000
Атаки систем SCADA	7	600/70/4000	800/60/3000	150/50/3500
Атаки на HMI	3	500/50/3000	400/60/3000	70/30/2600
Атаки підміна вузла («атака воронки»)	15	150/35/1500	100/55/1500	30/15/1500
Компрометація вузла збору даних	5	250/30/1700	190/35/1800	30/20/1300
Підміна маршрутизатора	11	300/40/2300	380/60/2500	35/20/1700
Знімання інформації з периферійних пристроїв	15	150/25/1500	75/20/1400	45/10/1000
Атаки на системи супутникової навігації	9	90/30/4000	150/50/4000	20/15/150

Примітка: * – За даними: [189, 200, 210]; ** – Ознаки та їх інформативність за даними: [137, 148, 162, 164, 210];
*** – За даними: [132, 151]; **** – За даними: [80, 114, 132, 170, 188].

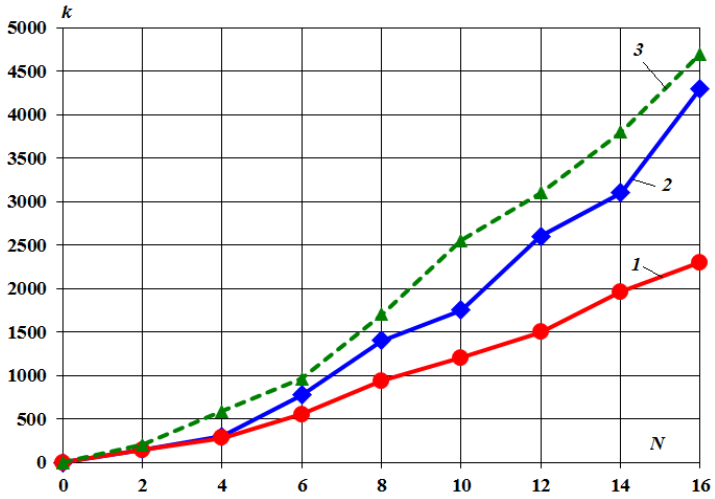
На рис. 4.11 показані результати тестування запропонованої для ЕС моделі для розпізнавання складних вірусних атак за ознаками у порівнянні з ефективністю вище згаданих методів та моделей [80, 137, 162, 210]. Прийняті на рис. 4.10 позначення, збережено.



1 – АСР; 2 – методи прогнозування станів; 3 – послідовний перебір ознак

Рис. 4.11. Порівняльна ефективність запропонованої моделі для розпізнавання складних вірусних атак за ознаками

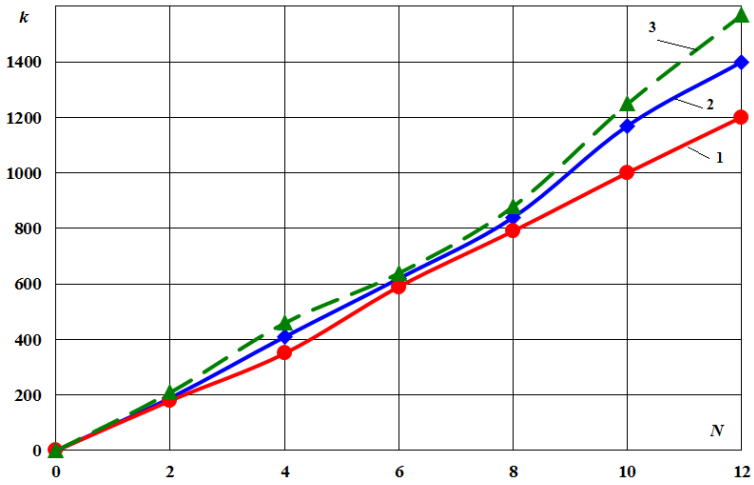
На рис. 4.12 показані результати тестування ЕС для розпізнавання атак на системи SCADA.



1 – АСР; 2 – методи прогнозування станів; 3 – послідовний перебір ознак

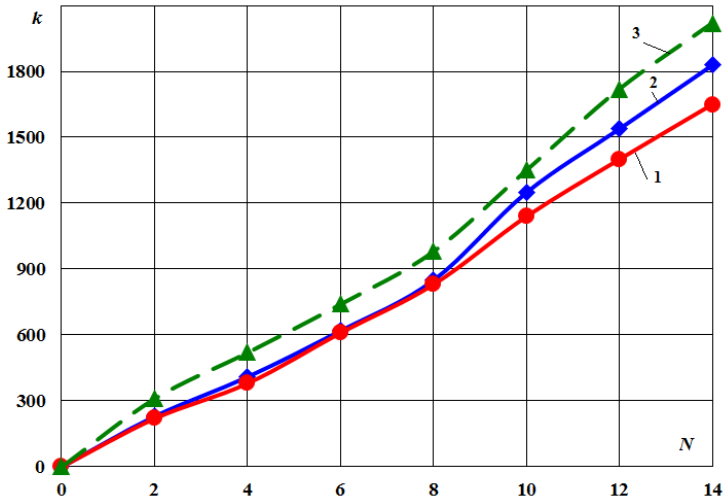
Рис. 4.12. Порівняльна ефективність запропонованої моделі для розпізнавання атак на системи SCADA

На рис. 4.13 та 4.14 показані результати тестування ЕС для розпізнавання атак на стандартні компоненти ПЗ КВІС (КВКС) та атак типу мережева розвідка, відповідно.



1 – АСР; 2 – методи прогнозування станів; 3 – послідовний перебір ознак

Рис. 4.13. Порівняльна ефективність запропонованої моделі для розпізнавання атак на стандартні компоненти ПЗ КВІС (КВКС)



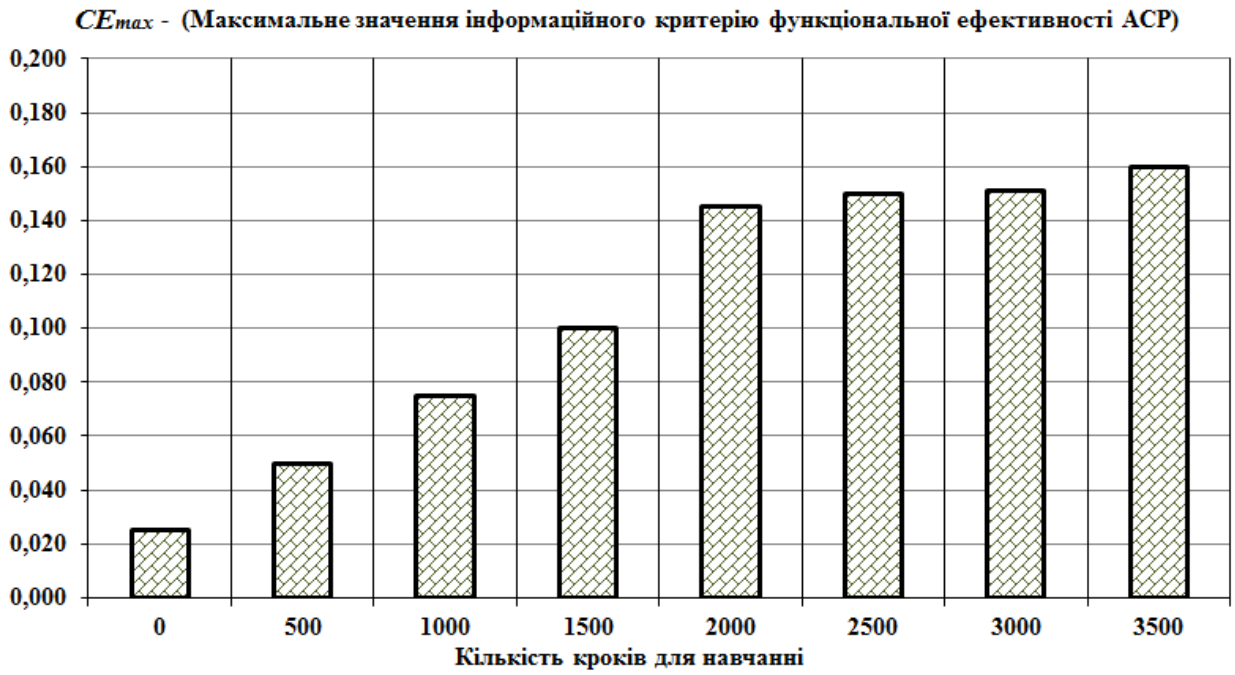
1 – АСР; 2 – методи прогнозування станів; 3 – послідовний перебір ознак

Рис. 4.14. Порівняльна ефективність запропонованої моделі для розпізнавання атак типу мережева розвідка

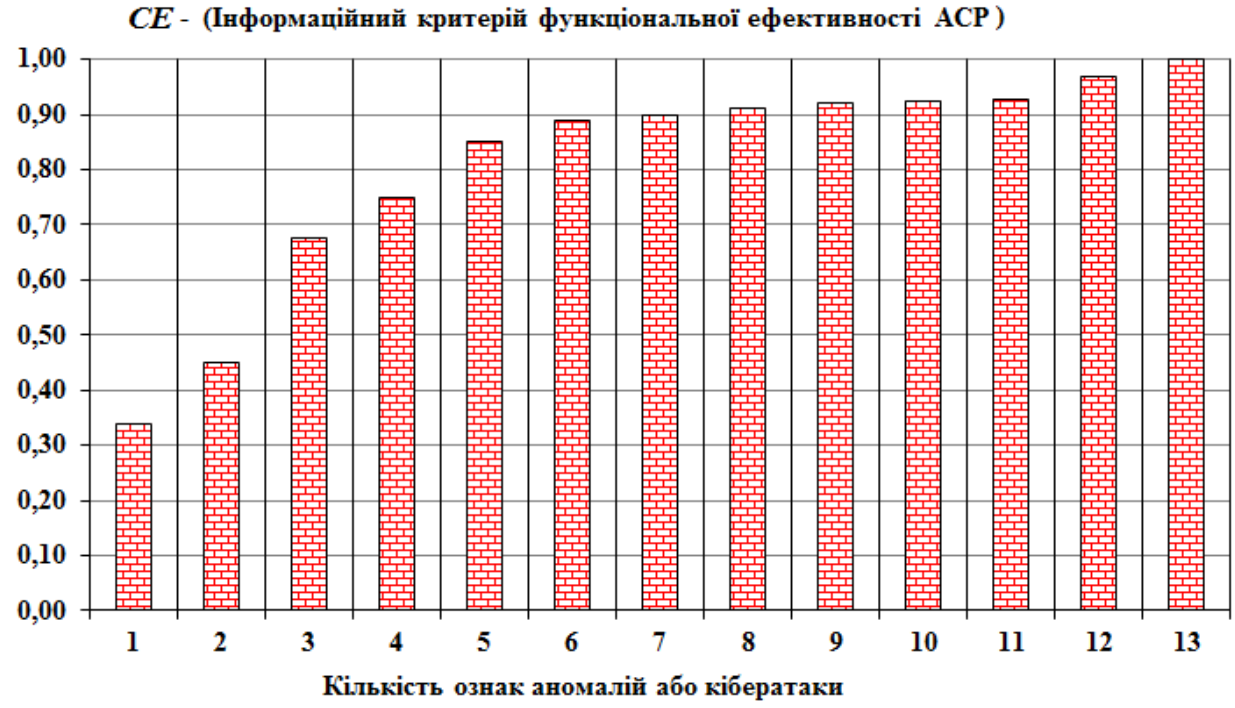
В порівнянні з широко вживаними в АСР методами послідовного перебору ознак і статистичними алгоритмами станів [80, 96, 114, 132, 141, 200, 211], використання методу, що базується на моделі інформаційно-екстремальної технології (МІЕТ), дозволяє скоротити обсяг необхідних правил розпізнавання об'єктів в рамках класу в 3–10 разів (в залежності від класу об'єктів – кібератаки), а, отже, істотно знизити час виявлення кібератак. У режимі тестового навчання АСР для запропонованої моделі раціональне число кроків навчання склало $k \approx 3000$ для відомих класів об'єктів рис. 4.15 а), й $k \approx 3500 \ldots 4500$ для більш складних кібератак. При цьому, інформаційний критерій функціональної ефективності СІРКЗ для 5–9 реалізацій ознак з можливістю корекції вирішальних правил, дорівнює $CE_c = 0{,}85 - 0{,}93$, рис. 4.15 б).

Складнощі навчання АСР (СІРКЗ або ЕС) з використанням апарату МІЕТ, пов'язані виключно з етапом отримання багатовимірної навчальної матриці ознак (об’єкту який використовується для навчання – ОВН) для кожного з класів який присутній у БЗ АСР або ЕС. Однак, розроблена МІЕТ, в порівнянні з результатами, отриманими для моделей, розглянутих у розділі 1, на основі кінцевих автоматів [156, 160], випадкового відбору [80], мереж Байєса [155, 211], нейронних мереж [110, 166, 192, 210], забезпечують суттєво меншу кількість потрібних ознак для класифікації загроз, скорочуючи при цьому час навчання адаптивної СІРКЗ. Крім того, розроблена ЕС «Analyzer of cyberthreats» здатна автоматично формувати розміри навчальної багатовимірної матриці ознак кібератак, не вимагаючи участі багатьох експертів.

Порівняльна характеристика найбільш поширених методів та моделей виявлення вторгнень наведена у таблиці 4.6. Як видно з представлених результатів, МІЕТ показує кращі результати у порівнянні з іншими методами розпізнавання. В таблиці 4.6 представлені середні результати по 4 класах мережевих атак.



а)



б)

Рис. 4.15. Залежності значення max ІУФР ($CE_{\max}$) для варіантів словників ознак кібератак від кількості кроків алгоритму навчання ЕС (а) та від кількості ознак які використовуються для навчання ЕС (б)

Таблиця 4.6

Порівняльна характеристика методів виявлення вторгнень

№ п/п	Джерело	Математичний апарат	Робота в умовах нечітких ознак атаки	База даних	Кількість вхідних параметрів	Пошук вторгнень та нормальної поведінки, %	Пошук нових вторгнень
1	Y. Yang, D. Jiang, M. Xia [137, 150, 199]	Ієрархічна самоорганізуюча карта	-	KDD- 99	41	Norm-96,4; DoS-96,2; U2R-37,1; R2L-43,1; Probe-94,3	-
2	H.F. Eid, A. Darwis, A.E. Hassanien, A. Abraham [161, 192, 200]	Метод опорних векторів	-			Norm-99,8; DoS-97,5; U2R-86,6; R2L-81,3; Probe-92,8	-
3	М.П. Комар, Д.І. Боднар, А.О. Саченко [166]	Нейрон Kohonen	-			Norm-97,2; DoS-98; U2R-30,8; R2L-36,5; Probe-92,8	-
4	H.Alipour, E. Khosrowsh, M.Esmaeili та ін. [205, 210]	Нейронний класифікатор	-			Norm-98,5; DoS-98,5; U2R-76,3; R2L-89; Probe-82,5	-
5	W. Sharafat, R. Naoum [182, 205]	Генетичний нейронний алгоритм	+			Norm-96,3; DoS-97,3; U2R-29,8; R2L-9,6 Probe-88,7	-
6	M. Abadeh, J. Habibi [161]	Гібридна нейрона мережа	+			Norm-96; DoS-98,8; U2R-72,8; R2L-33,45; Probe-86,2	-
7	Петренко Т.А.	Моделі розділів 2 та 3	+		10-12	Norm-98,7; DoS-99,1; U2R-76,5; R2L-90; Probe-84,2	+

Таким чином, запропонований підхід розпізнавання кібератак, заснований на MIET навчання ЕС, дозволяє підвищити рівень виявлення мережевих кібератак у КВІС. Виявлення деяких типів атак відбувається з ймовірністю 77–99 % при незначному рівні помилкових спрацювань, див. рис. 4.16, 4.17. Крім цього, запропонований метод не вимогливий до ресурсів ІС і здатний виявляти невідомі типи кібератак у КВІС.

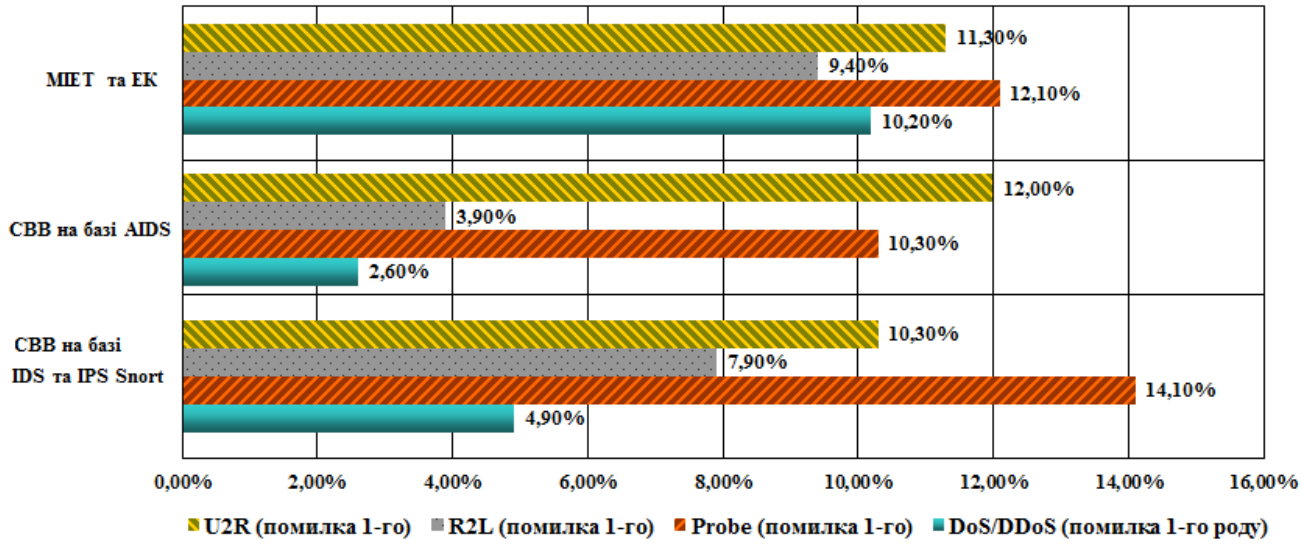


Рис. 4.16. Значення помилок першого роду при виявленні кібератак різними системами

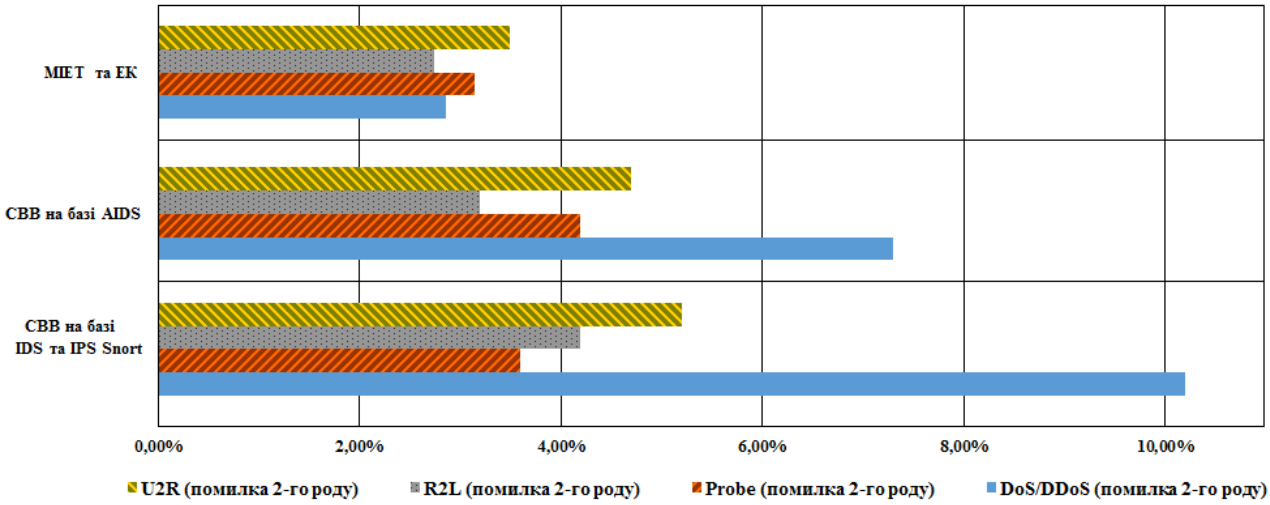


Рис. 4.17. Значення помилок другого роду при виявленні кібератак різними системами

У результаті означеного експерименту для розробленої МІЕТ та МІРК були отримані наступні результати для відповідних атак:

DoS/DDoS – для помилок 1-го роду (кількість помилкових спрацьовувань) – 10–11 %) і помилок 2-го роду (кількість невиявлених атак) – 2–3%;

Probe – для помилок 1-го роду – 12–14 і помилок 2-го роду – 3–4%;

R2L – для помилок 1-го роду – 9–11% і помилок 2-го роду – 2–4 %;

U2R – для помилок 1-го роду – 11–1 2% і помилок 2-го роду – 3–5%.

Дані результати дозволяють порівняти розроблені модель та алгоритм (МІЕТ) з дослідженими раніше в роботах [80, 96, 110, 166, 192, 211] методами та математичними моделями, які застосовуються у СВВ, рис. 4.18.

Як свідчать результати порівняльного аналізу, запропонована модель та алгоритм МІЕТ навчання ЕС «Analyzer of cyberthreats», дозволяють досягнути результатів розпізнавання типових класів кібератак на рівні від 76,5 % до 99,1%, що знаходиться на рівні ефективності розпізнавання гібридних нейронних мереж та генетичних алгоритмів.

Загроза зараження КВІС (КВКС) зловмисним програмним забезпеченням, наприклад, комп'ютерними вірусами, являє відчутну загрозу. Існуючі засоби захисту КВІС (КВКС) не завжди оперативно справляються з епідеміями комп'ютерних вірусів. Тому розробка та впровадження нових засобів виявлення та захисту, здатних розпізнати й запобігти зараженню комп’ютерних систем на ранніх стадіях є актуальною задачею для ЕС та СВВ. У розробленій ЕС «Analyzer of cyberthreats» як модель поширення комп'ютерних вірусів в КВІС (КВКС) була використана модель PSIDR [139], для прогнозування та оцінки процесу поширення складних вірусів в мережі Замовника.

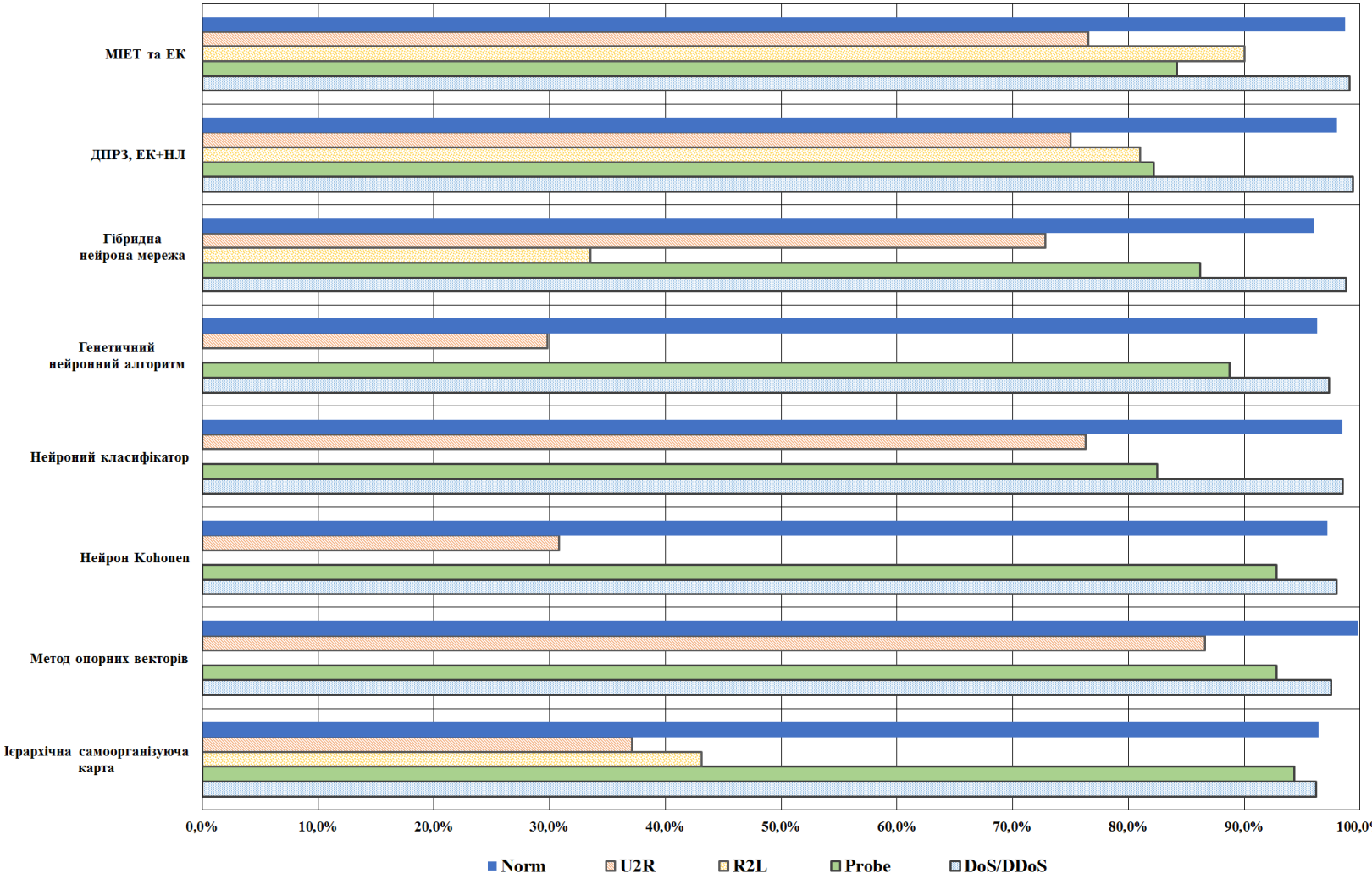


Рис. 4.18. Порівняльний аналіз ефективності різних математичних методів та моделей які використовуються у СРКА

Для тестування можливості СІРКЗ та ЕС «Analyzer of cyberthreats» виконувати завдання розпізнавання зараження КВІС вірусами, в ході досліджень були підготовлені програми – макети вірусів, див. рис. 4.19, 4.20. Програми–макети вірусів не становили небезпеки для комп'ютерної мережі підприємства Замовника, але поширювалися тими ж методами, що реалізовані в більшості сучасних шкідливих програм. Окрім того, макети вірусів характеризуються невеликою кількістю реалізацій ознак і відсутністю сигнатур у базах поширених антивірусних систем.

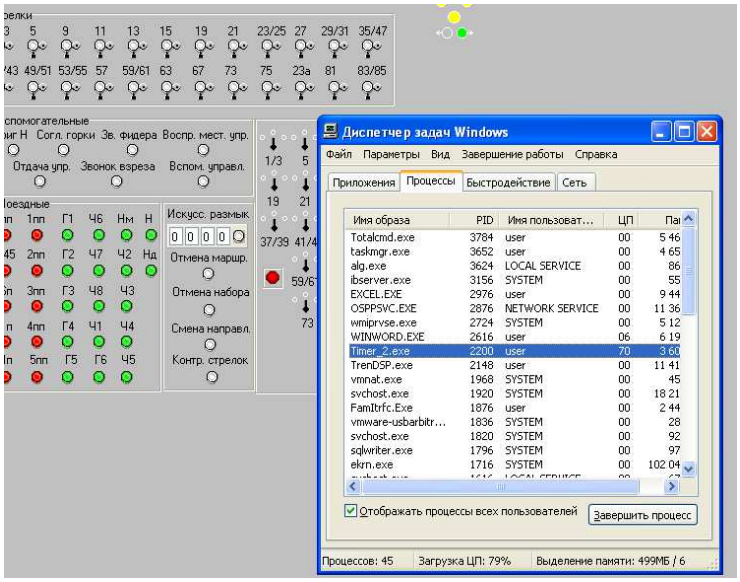


Рис. 4.19. Вплив макету вірусу на АРМ у складі АСК
(Завантаження АРМ: 70 % – 60 с)

При потраплянні програми – «вірусу» одним із перерахованих способів на АРМ і його подальшому запуску програма проводить такі дії:

1) знижується оцінка локальної антивірусної безпеки (тому що «вірус» уже був запущений);

2) проводиться пошук серед запущених процесів антивірусних програм. Якщо таких процесів немає, то оцінка локальної антивірусної безпеки знижується на кілька балів;

3) виконуються спроби доступу до розділів автозапуску і спеціалізованих папок користувача (наприклад, папка «Автозавантаження»). При успішному запису «вірусу» в автозапуск оцінка знижується;

4) проводиться спроба завантаження тестового файлу з локальної мережі. Успішне завантаження тестового файлу свідчить про відсутність або неправильне використання мережевого екрану, оцінка знижується.

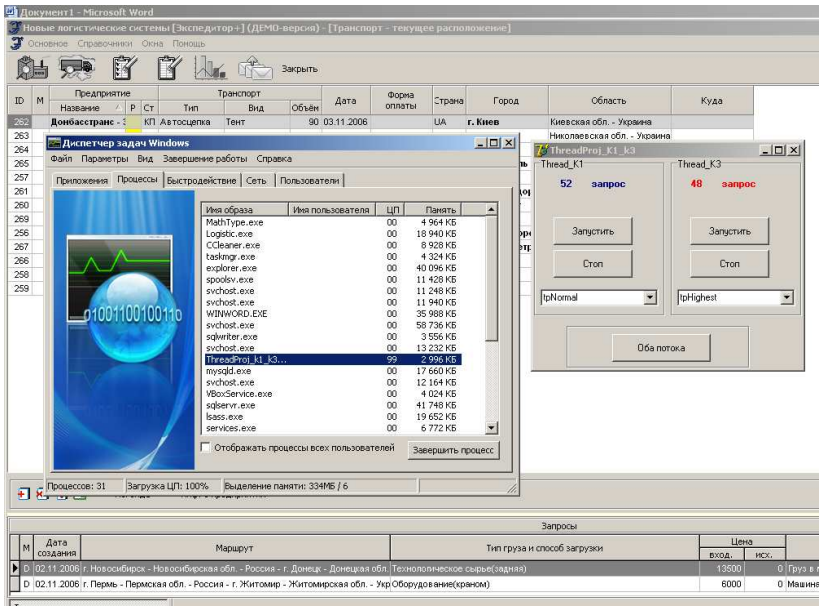


Рис. 4.20. Вплив макету вірусу на ІС «Експедитор»
(Завантаження ПК: 99 % – 30 с)

На основі експериментальних даних було отримано такий вираз, що характеризує залежність кількості «заражених» комп'ютерів від часу для розрахованих середніх оцінок безпеки комп'ютерної мережі підприємства:

$$x_{n+1} = 1 - (1 - P_{cp}) \cdot (1 - x_n)_n^{P_{cp} \cdot x}, \tag{4.6}$$

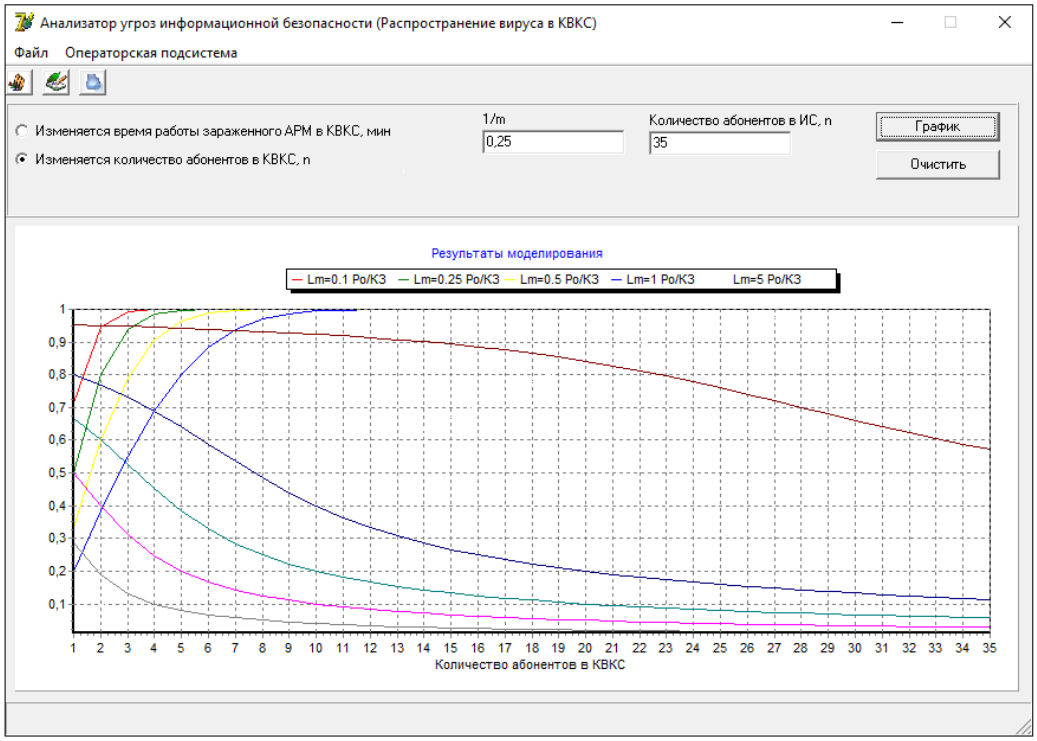
де P_{cp} – середня ймовірність зараження комп'ютерів, що обчислюється виходячи з різниць (*1 – M/10*), де *M* – оцінка локальної антивірусної безпеки комп'ютера за десятибальною шкалою, отримувана за допомогою аналізу

ряду чинників: наявності встановлених антивірусних програм у запущених процесах, наявності і якості роботи мережевого екрану, активності користувача в годинах роботи за комп'ютером, кількості спільних ресурсів і прав доступу до них, можливості автозапуску програм із зовнішніх носіїв інформації та частота їх використання, права доступу до параметрів автозавантаження системного реєстру.

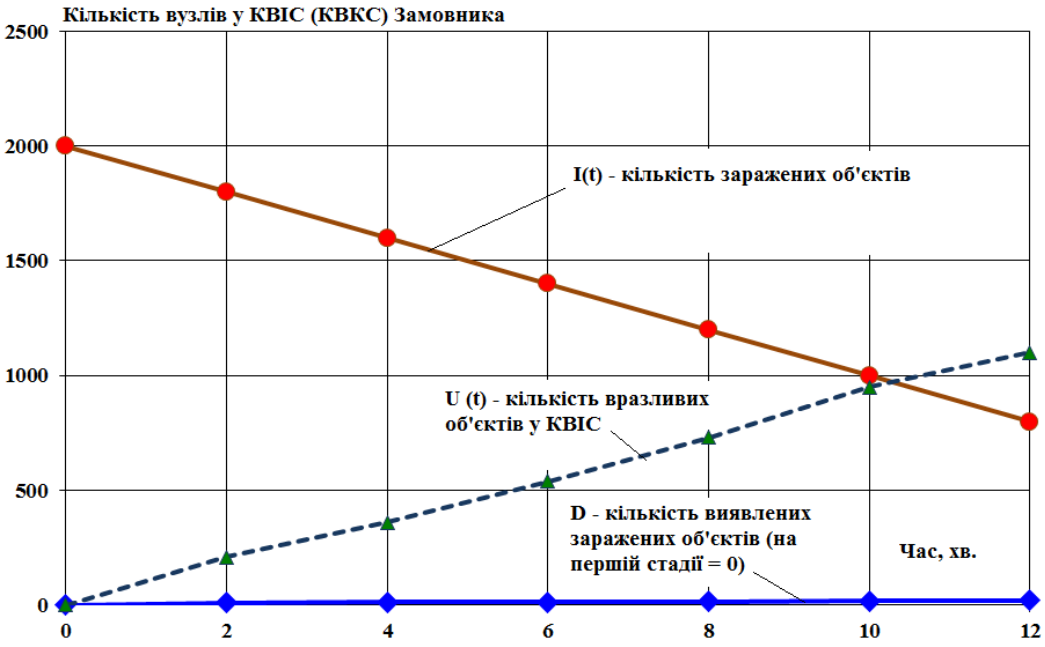
Розроблені «віруси» представляли собою клієнтські програми, які, «заразивши» АРМ у складі КВІС (КВКС), навантажували систему створеним процесом до 92–98%, збирали інформацію про рівень безпеки кінцевого ПК, наявність антивірусних програм, мережевих екранів і т. д. При цьому ЕС «Analyzer of cyberthreats» оцінювала кількість файлів на АРМ у складі КВІС (КВКС), які могли б бути «заражені» за час роботи «вірусу» до моменту виявлення і видалення. Всі дані під час дослідження надходили на сервер, де встановлена ЕС. Програма–сервер у складі ЕС встановлювалася на один із комп'ютерів мережі, що є первинним джерелом розповсюдження «вірусів».

В експерименті використовувалися методи поширення «вірусів» за допомогою копіювання в загальні папки або автозапуску з зовнішнього носія. Теоретично враховувалася можливість зараження потенційно небезпечних файлів за допомогою підрахунку їх кількості на комп'ютері: виконуваних (*.exe, *.com) і файлів, які підтримують запуск інших програм і скриптів (*.doc, *.docx, *.xls, *.xlsm, *.htm, *.bat, *.cmd). Також оцінювалася можливість доступу до розділів автозавантаження в системному реєстрі і спеціалізованих каталогів користувача.

На рис. 4.21 представлені отримані за допомогою ЕС «Analyzer of cyberthreats», рис. 4.21 а) графіки кривих залежності зміни кількості заражених, вразливих та виявлених заражених об’єктів на ЕОМ у складі КВІС Замовника, рис. 4.21 б).



а)



б)

Рис. 4.21. Залежність зміни кількості заражених, вразливих та виявлених заражених об’єктів на ЕОМ у складі КВІС Замовника

Проведений аналіз ефективності застосування модуля ЕС «Analyzer of cyberthreats» для завдання оцінки розповсюдження шкідливого ПЗ, показав, що розбиття моделі поширення комп'ютерних загроз на два етапи, дає можливість незалежного аналізу процесу зараження і лікування. Введення затримки між початком кожного з етапів, розпізнавання об’єкту в ЕС, його ідентифікації, локалізації та лікування, дозволило усунути один з недоліків антивірусних систем, які використовуються у КВІС Замовника – відсутність можливості пояснення складних випадків зміну стану системи, для яких є невелика кількість реалізацій ознак та відсутні сигнатури у базах антивірусних систем.

Таким чином, тестування ІС Замовника дозволило визначити поведінку всієї інфраструктури у звичайних умовах і при різних варіантах атак (DoS/DDoS, Probe, R2L, U2R, зараження АРМ вірусом та ін.).

Пошук уразливостей у програмному забезпеченні мережевих служб підприємства не дав позитивних результатів.

У ході тесту були зроблені контрольні запуски експлойтів на служби SMTP і WWW для перевірки можливого блокування IP-адреси, з якої проводився запуск, засобами міжмережевого екрану. Підсистема захисту не відреагувала на активні спроби атаки, адреса «порушника» заблокованою не була.

Аналіз результатів, отриманих під час тестування розробленої адаптивної здатної до самонавчання ЕС «Analyzer of cyberthreats» показує, що досить ефективним для поповнення БЗ є використання алгоритму із 5–10 ознаками ОВН, тобто для цього випадку ІУФР досягає свого максимального значення, що свідчить про побудову безпомилкових за навчальною матрицею вирішальних правил.

В таблиці 4.7 наведені результати експериментальних досліджень з інтелектуального розпізнавання кібератак у ІС та АСК підприємств Замовників, а також порівняння із результатами імітаційного моделювання, розглянутого у попередньому розділі роботи.

Таблиця 4.7

Результати експериментальних досліджень з інтелектуального розпізнавання кібератак у ІС підприємств у порівнянні імітаційними моделями

№	Параметр	Замовник №1	Замовник №2	Модель	Відхилення, %
1	Виявлено атак за типами, %: DoS/DDoS Probe R2L U2R	 92–98 88–95 49–50 67–70	 91–95 90–94 42–45 65–69	 85–96 – – –	7–9
2	Ймовірність досягнення зловмисником мети за $T_{зад}=60$ хвилин: DoS/DDoS Probe R2L U2R	 0,10–0,11 0,08–0,11 0,06–0,07 0,04–0,06	 0,06–0,092 0,05–0,067 0,02–0,04 0,02–0,05	0,08–0,1	0,05–0,1
3	Час отриманням доступу незареєстрованого користувача до комп'ютера з боку віддаленої машини (атака R2L), год	4–5	7–9	-	-
4	Час отриманням зареєстрованим користувачем права адміністратора (атака U2R), год	1–1,5	1–2	-	-

До проведення тестування на проникнення не варто підходити як до завдання, що виконується один раз. В умовах функціонування сучасних підприємств ІТ-інфраструктура може дуже швидко розвиватися і відповідно змінюватися. Змінюються властивості мережі, склад і якість прикладних сервісів, число користувачів тощо. Тому оцінка реакції КВІС на підвищення навантаження повинна бути регулярною та виконуватися, наприклад, при проведенні періодичного аудиту ІБ.

4.3. Висновки до четвертого розділу

Результатом проведених у даному розділі досліджень стали такі висновки:

1) Проведено тести на проникнення (за погодженням з керівництвом) в ІС та АСК підприємств Замовників, та виконано дослідження продуктивності ЛОМ і ІС, а також поведінки всієї інфраструктури в звичайних умовах і при різних варіантах кібератак (DoS/DDoS, Probe, R2L, U2R). Апробована методологія тестування адаптивної ЕС «Analyzer of cyberthreats», яка дозволяє на етапі аналізу та синтезу СІРКЗ для конкретних КВІС, встановлювати відношення між елементами адаптивних систем кіберзахисту;

2) Експериментально підтверджена ефективність запропонованого алгоритму навчання ЕС та СІРКЗ з можливістю корекції вирішальних правил, що дозволило створити адаптивний механізм самонавчання системи розпізнавання кібератак на КВІС;

3) Протестована розроблена ЕС «Analyzer of cyberthreats», яка дозволяє вирішувати такі завдання: розпізнавання загроз ІБ для КВІС та КВКС, збір інформації про стан ЕОМ у мережі підприємства; сканування запущених програм і процесів на АРМ підприємства; сканування доступних портів; визначення рівнів безпеки комп'ютера (АРМ) і можливість подальшого поширення «вірусу» з даного АРМ; фіксації часу «зараження» комп'ютерів у мережі; оцінка поточних ризиків НСД до КВІС підприємства та ін.;

4) Отримано наступні результати (для означеного експерименту в ході тестування розробленої моделі інтелектуального розпізнавання кібератак): DoS/DDoS – для помилок 1-го роду (кількість помилкових спрацьовувань) – 10–11 %) і помилок 2-го роду (кількість невиявлених атак) – 2–3%; Probe – для помилок 1-го роду – 12–14 і помилок 2-го роду – 3–4%; для атак R2L – для помилок 1-го роду – 9–11% і помилок 2-го роду – 2–4 %; U2R – для помилок 1-го роду – 11–1 2% і помилок 2-го роду – 3–5%.

5) Встановлено, що запропонований алгоритм навчання ЕС «Analyzer of cyberthreats» та відповідного поповнення БЗ для СІРКЗ, є найбільш ефективним для 3 кластерів в завданнях розбиття простору ознак атак для тестуємих КВІС. При цьому, в режимі тестового навчання ЕС та СІРКЗ достатня кількість кроків для безпомилкового визначення класів кібератак склала $k = 2500 - 3000$;

6) Виконано порівняння розроблених моделі та алгоритму навчання ЕС (МІЕТ) з існуючими, та доведено, що запропонована модель та алгоритм МІЕТ навчання ЕС «Analyzer of cyberthreats», дозволяють досягнути результатів розпізнавання типових класів кібератак на рівні від 76,5 % до 99,1%, що знаходиться на рівні ефективності розпізнавання гібридних нейронних мереж та генетичних алгоритмів;

7) Виявлено, що серед загроз ІБ підприємств Замовників на яких проводився тест на проникнення, найбільш суттєвими є: використовування застарілого ПЗ; адміністративні та технологічні труднощі для оновлення ПЗ та СУБД; доступ сторонніх компаній до технологічних мереж. В системах SCADA, загрозами ІБ є поширені уразливості ПЗ до атак DoS/DDoS та SQL Injection.

ВИСНОВКИ

У дисертації запропонований новий підхід до розв'язання актуальної науково-прикладної задачі підвищення ефективності систем інтелектуального розпізнавання кібератак на критично важливі інформаційні системи на основі розроблених моделей та методології створення здатної до самонавчання експертної системи, яка дозволяє враховувати відомі статистичні параметри кластеризації реалізацій ознак кібератак. Це дозволяє оперативно виявляти нові види складних комбінованих атак при обмежених обчислювальних ресурсах та варіативності умов застосування. Проведені дослідження дозволяють зробити такі висновки:

1. З’ясовано, що складність застосування до інтелектуальних систем розпізнавання цільових кібератак формалізованого апарату аналізу й синтезу СІРКЗ, полягає в тому, що конкретний інформаційний комплекс КВІС або КВКС та їх підсистеми ІБ складаються з різнорідних елементів, які описуються із використанням різних моделей. Показано, що застосування елементів адаптивного захисту інформації може бути засноване на використанні новітніх методів інтелектуального розпізнавання кібератак.

2. Запропонована модель ЕС у складі СІРКЗ із використанням процедури нечіткої кластеризації реалізацій ознак кібератак та можливістю корекції вирішальних правил, що дозволить створювати адаптивні механізми самонавчання системи інтелектуального розпізнавання кібератак на КВІС.

3. Запропоновано для оцінки якості розбиття простору реалізацій ознак об’єктів розпізнавання у ЕС застосовувати в якості оціночного показника модифіковану ІУФР, здатної до самонавчання системи розпізнавання. Доведено, що застосування моделі та методу кластерізації реалізацій ознак ОР, які ґрунтуються на ентропійному та інформаційно-дистанційному критерії Кульбака – Лейблера, дозволяє отримувати вхідну навчальну матрицю, яка використовується як об’єкт навчання, та в рамках

інтелектуальних технологій та методів навчання АСР будувати коректні вирішальні правила розпізнавання кібератак на КВІС. Встановлено, що збільшення кількості векторів–реалізацій класів ОР при виявленні кібератак проти КВІС призводить до збільшення максимального значення інформаційної умови функціональної результативності, а також дозволяє отримувати коректні правила для здатної до самонавчання системи розпізнавання. Доведено, що оцінка якості розбиття простору реалізацій ознак кібератак та інших варіантів легітимного втручання в роботу КВІС, може бути ефективно вирішена на основі ІУФР та можливої корекції вирішальних правил розпізнавання, що дозволяє зменшити кількість попередньої інформації, яка підлягає опрацюванню аналітиками служб ІБ КВІС.

4. Удосконалено метод навчання ЕС у складі СІРКЗ для запропонованих етапів паралельної оптимізації контрольних відхилень для реалізацій ознак розпізнавання кібератак проти КВІС. Розроблено метод навчання ЕС у складі СІРКЗ, який являє собою ітераційну процедуру пошуку глобального максимуму ІУФР у допустимому діапазоні визначення його функції. Запропоновані уточнення до методу навчання ЕС дозволяють попереджати можливі випадки поглинання одним класом ОР базових реалізацій ознак іншого класу, враховує відомі статистичні параметри кластеризації реалізацій ознак об’єктів розпізнавання, а також помилки під час завдання на прийняття рішення в ході процедур машинного навчання. Отримано відповідні предикатні вирази для ЕС здатної до самонавчання.

5. Запропоновано уточнення до методу розбиття простору реалізацій ознак на кластери в ході реалізації процедури розпізнавання ОР, який відрізняється від існуючих одночасною оптимізацією при обчисленні контрольних допусків у ході аналізу ЕС реалізацій ознак ОР та дозволяє на кожному кроці навчання змінювати перевірочні допустимі відхилення для всіх ознак одночасно.

6. Проведено дослідження за допомогою імітаційного моделювання для

моделі та методу кластерізації реалізацій ознак ОР, які ґрунтуються на ентропійному та інформаційно-дистанційному критерії Кульбака – Лейблера, а також алгоритму оптимізації контрольних допусків на реалізації ознак розпізнавання ОР. Встановлено, що квазіоптимальне значення параметра системи контрольних допустимих відхилень дорівнює 8–16 % при максимальному значенні ІУФР $CE_{\max} = 6{,}16$.

7. Проведено тести на проникнення в ІС та АСК підприємств Замовників, та виконано дослідження продуктивності ЛОМ і ІС, а також поведінки всієї інфраструктури в звичайних умовах і при різних варіантах кібератак (DoS/DDoS, Probe, R2L, U2R). Апробована методологія тестування ЕС «Analyzer of cyberthreats», яка дозволяє на етапі аналізу та синтезу СІРКЗ для конкретних КВІС, встановлювати відношення між елементами адаптивних систем кіберзахисту. Експериментально підтверджена ефективність запропонованого методу навчання ЕС та СІРКЗ з можливістю корекції вирішальних правил, що дозволило створити адаптивний механізм самонавчання системи розпізнавання кібератак на КВІС. Протестована розроблена ЕС «Analyzer of cyberthreats». Встановлено, що запропонований метод навчання ЕС «Analyzer of cyberthreats» та відповідного поповнення БЗ для СІРКЗ, є найбільш ефективним для 3 кластерів у завданнях розбиття простору реалізацій ознак ОР для досліджених КВІС. При цьому в режимі тестового навчання ЕС та СІРКЗ достатня кількість кроків для безпомилкового визначення класів кібератак склала $k = 2500 - 3000$.

8. Виконано порівняння розроблених методів навчання (МІЕТ) з існуючими та доведено, що запропоновані рішення, дозволяють досягнути результатів розпізнавання типових класів кібератак на рівні 70 – 99%, що знаходиться на рівні ефективності розпізнавання гібридних нейронних мереж та генетичних алгоритмів.

Мета дисертаційного дослідження досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Азарсков В.М., Гізун А.І., Грехов А.М. і Скворцов С.О. «Параметри прогнозування і ідентифікації атак в інформаційно-комунікаційних системах», *Захист інформації*, т. 16, № 1, с. 89–93, 2014.

2. Алексеев А.С. и Котенко И.В. «Командная работа агентов по защите от распределенных атак “отказ в обслуживании», С*б. докл. VI Международной конф. по мягким вычислениям и измерениям SCM’2003,* т.1, с. 294–297, 2003.

3. Алексеев В.М., Андрианов В.В., Зефиров С.Л. и Лупанов И.Ю. «Комплекс защиты информации для автоматизированных информационных систем с использованием баз данных» *Безопасность информационных технологий*, № 4. с. 128–130, 1994.

4. Алипов И.Н. «Методы защиты информации при ее передаче» : дис. канд. техн. наук: 05.13.08 Харьков, с.171, 1997.

5. Андерсон Т. «Введение в многомерный статистический аналіз»,: пер. с англ. *М. Физматлиз*, с. 400, 1963.

6. Анин Б.Ю. «Защита компьютерной информации» *СПб. : Петербург*, с. 384 2000.

7. Бабенко Т.В. «Дослідження ентропії мережевого трафіка як індикатора DDOS-атак», *Науковий вісник НГУ*, № 2, с. 86-89, 2013.

8. Барсуков В.С. и Водолазкий В.В., «Современные технологии безопасности: Интегральный подход», *М. : Нолидж*, с. 496, 2000.

9. Баскакова Л.В. и Журавлев Ю.И «Модель распознающих алгоритмов с представительными наборами и системами опорных множеств», *Журн. Выч. матем. и матем. физики*, № 5. с. 1264–1275, 1981.

10. Бачило И.Л, Лопатин В.Н. и Федотов М.А. «Информационное право», *СПб.: «Юридический центр Пресс», с.* 246, 2001.

11. Безкоровайный М.М., Костогрызов А.И. и Львов В.М. «Инструментально-моделирующий комплекс оценки качества

функционирования информационных систем», *М. : Вооружение. Политика. Конверсия*, с.305, 2002.

12. Безпека авіації / В.П. Бабак [та ін.]. Київ : *Техніка*, 2004. – 584 с.

13. Биктимиров М.Р. и Щербаков А.Ю. «Избранные главы компьютерной безопасности», *Казань: Изд-во Казан. матем. об-ва*, с. 372, 2004.

14. Бірюков Д.С. і. Кондратов С.І. «Захист критичної інфраструктури: проблеми та перспективи впровадження в Україні», *Київ: НІСД*, с. 96, 2012.

15. Бородин А.В. «Теоретико-игровые модели процессов риска над сетями Петри», «Моделирование и анализ безопасности и риска в сложных системах», *Труды международной научной школы МАБР,* с. 305-307, 2006.

16. Бородюк В.П. и Львова А.В. «Повышение экономической эффективности системы информационной безопасности», *Вестник МЭИ*, № 4, с. 89–92, 2007.

17. Бочков М.В., Логинов В.А. и Саенко И.Б. «Активный аудит действий пользователей в защищенной сети», *Защита информации. Конфидент*, № 45, с. 94–98, 2002.

18. Брюхомицкий Ю.А. и Казарин М.Н. «Метод обучения нейросетевых биометрических систем на основе построения аппроксимированных областей», *Известия ТРТУ*, № 4, с. 155–159, 2007

19. Булах Е.В. «Конечные автоматы с псевдослучайными переходами и методы защиты информации на их основе», дис. канд. техн. наук, 05.13.13 – Харьков, с. 164, 2003.

20. Бурячок В.Л. «Застосування бездротових мереж в ході організації та проведення розвідки систем телекомунікацій», *Захист інформації*, т. 15, № 4, с. 284–291, 2013.

21. Вайнцвайг М.Н. «Алгоритм обучения распознавания образов «Кора»», *М. : Сов. радио*,с. 82–91, 1973.

22. Васильев В.Г., Кузьмук В.В. и Петри С. «Параллельные алгоритмы и модели мультипроцессорных систем», *Киев : Наук. думка*, с. 212, 1990.

23. Василіу Є. «Послідовна атака пасивного перехоплення двох зловмисників на пінг-понг протокол з ГХЦ-триплетами кубітів», *Захист інформації*, т. 16, № 2. с. 97–105. 2014.

24. Волковский Н.Л. «История информационных войн», *СПб. : Полигон*, с. 502. 2003.

25. Волянська В.В., Корченко А.О. та Паціра Є.В. «Система виявлення аномалій на основі нечітких моделей», *Зб. наук. пр. Інституту проблем моделювання в енергетиці НАН України ім. Г. Є Пухова*. т.2, с. 56–60, 2007.

26. Востоцький В.О. та Занченко С.А. «Інформаційно-екстремальний алгоритм агломеративного кластер-аналізу», *Вісник Сумського державного університету. Серія технічні науки*, № 3, с. 107–114, 2012.

27. Вульфин А.М., Гиниятуллин В.М. и Фрид А.И. «Нейросетевая модель выявления и распознавания технологических ситуаций в рамках методологи Data Mining», *Нейроинформатика*, № 1, с. 132–142, 2010.

28. Вятченин Д.А. «Нечеткие методы автоматической классификации», Монография. *Минск : УП "Технопринт*, с. 219, 2004.

29. Гайкович В.Ю. и Ершов Д.В. «Основы безопасности информационных технологий», *М.: МИФИ*, с. 86, 1995.

30. Галатенко В.А. «Оценка безопасности автоматизированных систем», *Jet info*, № 7 (146), с. 16 –21. 2005.

31. Галицкий А.В., Рябко С.Д. и Шаньгин В.Ф. «Защита информации в сети - анализ технологий и синтез решений», *М. : ДМК Пресс*, с. 616, 2004.

32. Герасименко В.А. и Малюк А.А. «Защита информации в автоматизированных системах обработки даннях», *М. : Энергоатомиздат*, с. 512, 1997.

33. Глова В.И. и Насыров Р.И.«Показатели качества функционирования и стратегии обслуживания систем защиты информации», *Вестник КГТУ им. А.Н. Туполева*, №4, с. 39–43, 2006.

34. Глушаков С.В., Хачаров Т.С. и Соболев Р.О. «Секреты хакера: защита и атака», *Харьков : Фолио*, с. 414, 2008.

35. Гопиенко А.В., Куц Ю.В. и Монченко Е.В. «Метод скрытой передачи данных в компьютеризированных информационно-измерительных системах», *Захист інформації*, № 2, с. 107–111, 2011.

36. Горбунов В.А. «Математические методы в теории защиты информации», *М. : Изд-во МГТУ*, с. 82, 2004.

37. Городецкий В.И. и Котенко И.В. «Концептуальные основы стохастического моделирования в среде Интернет», *Труды института системного анализа РАН,* т. 9, с. 168–185, 2005.

38. Грездов Г.Г. «Модифицированный способ решения задачи формирования эффективной комплексной системы защиты информации автоматизированной системы», *Киев : ГУИКТ*, с. 132, 2009.

39. Грездов Г.Г. «Способ решения задачи формирования комплексной системы защиты информации для автоматизированных систем 1 и 2 класса», *К.*, с. 64, 2005.

40. Гриняев С. «Концепция ведения информационной войны в некоторых странах мира», *Зарубежное военное обозрение*, № 2, с. 11–15, 2002.

41. Грищук Р., Охрімчук В. і Ахтирцева В. «Джерела первинних даних для розроблення шаблонів потенційно небезпечних кібератак», *Захист інформації*, т. 18, № 1, с. 21–29, 2016.

42. Грищук Р.В. і Мамарєв В.М. «Метод оцінювання інформативності параметрів потоку вхідних даних для мережевих систем виявлення атак», *Системи обробки інформації*, в. 4(1), с. 103-107, 2012.

43. Грищук Р. В. Теоретичні основи моделювання процесів нападу на інформацію методами теорій диференціальних ігор та диференціальних перетворень : монографія, Житомир : РУТА, 2010. – 280 с.

44. Грушо А.А. и Тимонина Е.Е. «Теоретические основы защиты информации», *М. : Яхтсмен*, с. 67, 1996.

45. Давиденко А.М., Головань С.М. і Щербак Л.М. «Структурована база загроз для інформації в інформаційних системах», *Моделювання та інформаційні технології*, в. 32, с. 17–22, 2006.

46. Давиденко А.М., Головань С.М. і Щербак Л.М. «Аналіз дій загроз у автоматизованих системах обробки інформації», *Моделювання та інформаційні технології*, № 36, с. 3–8. 2006.

47. Давиденко А.М. і Суліма О. А. «Використання формальних засобів опису процесів надання повноважень», *Захист інформації*, т. 18, № 2, с, 143–149, 2016.

48. Джеймс Л. «Фишинг. Техника компьютерных пре ступлений», *М. : НТ Пресс*, с. 320, 2008.

49. Дмитриев А.И., Журавлев Ю.И. и Кренделев Ю.И. «О математических принципах классификации предметов и явлений», *Дискретный аналіз*, в. 7, с. 1–17, 1966.

50. Довбыш А.С., Будник Н.Н. и Москаленко В.В. «Информационно-экстремальный алгоритм оптимизации параметров гиперэллипсоидных контейнеров классов распознавания», *Международный научно-технический журнал «Проблемы управления и информатики»*, № 5, с. 111–119, 2012.

51. Довбиш А.С. і Мартиненко С.С. «Інформаційно-екстремальний метод розпізнавання електронограм», №2, с.85–92, 2009.

52. Довбиш А.С. «Основи проектування інтелектуальних систем», *Навч. посіб., Суми : СумДУ*, с. 170, 2009.

53. Домарев В.В. «Защита информации и и безопасность компьютерных систем». *Київ : Издательство Диасофт*, с. 480, 1999.

54. Дьяконов В.П. и Круглов В.В. «MATLAB. Анализ, идентификация и моделирование систем», *Специальный справочник. СПб. : ПИТЕР*, с. 576, 2002.

55. Закон України “Про державну таємницю”, *Відомості ВРУ*, 1999. № 49. Ст. 428.

56. Закон України “Про захист інформації в інформаційно-телекомунікаційних системах”, *Відомості ВРУ*, 2004. № 31. Ст. 286.

57. Защита информации в телекоммуникационных системах / Конахович Г.Ф. [и др.], *Киев : "МК-Пресс"*, 2005. 288 с.

58. Защита от хакеров беспроводных сетей / Барнс К. [и др.], *М.: ДМК Пресс*, 239 с., 2005.

59. Защита от хакеров корпоративных сетей / Линч У. [и др.], *М.: ДМК Пресс,* 863 с., 2005.

60. Зегжда Д.П. и Ивашко А.М. «Основы безопасности информационных». *М. : Горячая линия – Телеком*, 452 с, 2000.

61. Ивахненко А.Г., Савченко Е.А., Ивахненко Г.А. и Синявский В.Л. «Проблемы индуктивного двухуровневого мониторинга сложных процес сов», *Управляющие системы и машины*, № 3, с. 13–21, 2007.

62. Карпинский Н.П., Корченко А.А. і Ахметова С.Т. «Метод формування базових детекційних правил для систем виявлення вторгнень», *Захист інформації*, т. 17, № 4, с. 312–324, 2015.

63. Кеммерер Р. и Виджна Дж. «Обнаружение вторжений: краткая история и обзор», *Открытые системы*, № 7, с. 8–15, 2002.

64. Козиол Дж. «Искусство взлома и защиты систем», *СПб. : Питер*, с.416, 2006.

65. Комаров А.А. «Тесты на проникновение: методики и современные подходы», *Журнал «IT-спец»*, № 2, с. 48–53, 2009.

66. Коржик В.И. и Кушнир Д.В. «Теоретические основы информационной безопасности телекоммуникационных систем», *СПб. : СПбГУТ*, с.134, 2000.

67. Корнеев В.В. и Маслович А.И. «Распознавание программных модулей и обнаружение несанкционированных действий с применением аппарата нейросетей», *Информационные технологи*, № 10, с. 34–48, 1997.

68. Корнеев Н.Р. и Беляев А.В. «Информационная безопасность предприятия», *СПб. : БХВ – Петербург*, с.752, 2003.

69. Корниенко Б.Я. и Щербак Л.Н. «Анализ действия и методов противодействия информационным угрозам типа "Riskware"», *Захист інформації,* № 1, с. 54–59, 2008.

70. Корченко А.А. «Модель эвристических правил на логико-лингвистических связках для обнаружения аномалий в компьютерных системах», *Захист інформації*, № 4 (57), с. 112-118, 2012.

71. Корченко О.Г., Терейковський І.А. і Дзюбаненко А.О. «Сучасні нейромережеві методи та моделі оцінки параметрів безпеки ресурсів інформаційних систем», *Захист інформації*, т. 16, № 3, с. 223–232, 2014.

72. Костров Д. «Системы обнаружения атак», *Byte*, №8, с. 20–26, 2002.

73. Котенко И.В. и Степашкин М.В. «Интеллектуальная система моделирования атак на web-сервер для анализа уязвимостей компьютерных систем», *Сб. докл. VI Межд. конф. по мягким вычислениям и измерениям SCM'2003,* т. 1, с. 298–301, 2003.

74. Котенко И.В. и Юсупов Р.М. «Перспективные направления исследований в области компьютерной безопасности», *Защита информации. INSIDE*, № 2, с. 46–57, 2006.

75. Котенко И.В. «Модели противоборства команд агентов по реализации и защите от распределенных атак «Отказ в обслуживании»», *Тр. межд. научно-технич. конф. IEEE AIS'03 и CAD-2003*, т. 1, с. 422–428, 2003.

76. Коэн Ф. «50 способов обойти систему обнаружения атак», *URL: http://infosec.ru/pub/pub/13_09.htm.*

77. Куссуль Н.Н., Скакун С.В. и Лобунец А.Г. «Реализация нейросетевой модели пользователей компьютерных систем на основе агентной технологи», *Проблемы управления и інформатики*, № 2, с. 93-102, 2005.

78. Лахно В.А. і Петров О.С. «Моделювання інформаційної безпеки корпоративних систем підприємств з використанням теорії ігор і марківських процесів», *Вісник Національного університету "Львівська політехніка",* № 663, с. 122–127, 2009.

79. Лахно В.А., Петренко Т.А. і Пирог М.В. «Моделювання роботи адаптивної системи розпізнавання кібератак в умовах неоднорідних потоків запитів в модулях e-business», *Безпека інформації*, №2, с. 135–142, 2016.

80. Лахно В.А. «Побудова адаптивної системи розпізнавання кіберзагроз на основі нечіткої кластеризації ознак», *Східно-Європейський журнал передових технологій*, № 2/9 (80), с. 18–25, 2016.

81. Левин М. «Энцеклопедия начинающего хакера». *М. : «Новый издательский дом»*, с.1120, 2014.

82. Лейбин В.М. «Глобалистика, информатизация, системные исследования», *М. : Фаир-пресс*, с.280, 2007.

83. Ленков С.В., Перегудов Д.А. и Хорошко В.А. «Методы и средства защиты информации». *В 2-х томах. К.: Арий*, 2008.

84. Логинов В.А. «Методика активного аудита действий субъектов доступа в корпоративных вычислительных сетях на основе аппарата нечетких множеств», *Сб. докл. VI Междунар. конф. SCM'2003*, т. 1, с. 240–243, 2003.

85. Лукацкий А.В. «Информационная безопасность предприятий транспорта», *Решения Cisco Systems, T_Comm*, № 1_2, 2007.

86. Лукацкий А.В. «Обнаружение атак». *СПб : BHV*, с. 611, 2001.

87. Луценко Е.В. «Интеллектуальные информационные системы», *Краснодар : КубГАУ*, с.615, 2006.

88. Луцкий М.Г., Корченко А.А., Гавриленко А.В. и Охрименко А.А. «Модели эталонов лингвистических переменных для систем выявления атак», *Захист інформації*, №2 (55), с. 71–78, 2012.

89. Мамаев М. и Петренко С. «Технологии защиты информации в Интернете», *Спец. справ. СПб. : Питер*, с.844, 2002.

90. Манойло А.В., Петренко Л.И. и Фролов Д.Б. «Государственная информационная политика в условиях информационно-психологической войны», *М. : Горячая линия–Телеком*, с.541, 2007.

91. Медведский И.Д., Семьянов П.В. и Леонов Д.Г. «Атака на Интернет». *М. ДМК*, с.128, 1999.

92. Мельников В.П. «Безопасность информации в автоматизированных системах», *М. : Финансы и статистика*, с.368, 2003.

93. Мехед Д.Б., Ткач Ю.М., Базилевиич В.М. і Петренко Т.А. «Аналіз загроз інформаційної безпеки в мережах стандарту IEEE 802.11», *Захист інформації*, т. 17, № 4, с. 285–291, 2015.

94. Москаленко В.В. «Інформаційно-екстремальне навчання системи підтримки прийняття рішень з адаптивною кластеризацією даних», *Вісник СумДУ, Серія "Технічні науки"*, №3, с. 92–106, 2012.

95. Нормативне забезпечення інформаційної безпеки / [за ред. проф. В.О. Хорошка]. *Київ, ДУІКТ*, 2008. 533 с.

96. Норткат С. «Анализ типовых нарушений безопасности в сетях». *М., «Вильямс»*, с. 424, 2006.

97. Оксіюк О.Г. «Методика розрахунку часу затримки інформації управління в інформаційно-комунікаційних мережах», *Вісник ЧДТУ. Серія: Технічні науки*, № 3, с. 133–140, 2015.

98. Осипов В.Ю. «Концептуальные положения программного подавления вычислительных систем», *Защита информации. Конфидент*, № 4–5, с. 89–93, 2002.

99. Петренко Т.А. «Інформаційна безпека в сучасних умовах», *Вісник Чернігівського державного інституту права, соціальних технологій та праці*. №2. с. 98–102, 2009.

100. Петров А.С., Лахно В.А. и Ленков А.С. «Вероятностные модели конфликтных потоков данных в системах защиты информации корпоративных сетей», *Збірник наукових праць Військового інституту КНУ імені Тараса Шевченка*, №22. с. 99–107, 2009.

101. Поповский В.В. и Перенков А.В. «Защита информации в телекоммуникационных системах», *В 2-х т. Харьков : ООО «Компания СМИТ»*, с. 292, 2006.

102. Рогов П.Д., Малахов М.А. і Бухало Л.В. «Методика визначення рівнів загроз інформаційній безпеці держави у воєнній сфері», *Збірник наукових праць Військового інституту КНУ імені Тараса Шевченка*, № 39, с.143–147, 2013.

103. Романец Ю.В., Тимофеев П.А. и Шаньгин В.Ф. «Защита информации в компьютерных системах и сетях», *М. : Радио и связь*, с. 376, 2001.

104. Смирнов А.А., Дрейс Ю.А. и Даниленко Д.А. «Статистические свойства трафика на основе BDS-тестов для реализации системы обнаружения и предотвращения вторжений в телекоммуникационные сети», *Захист інформації.* т. 16, №2, С, 158–167, 2014.

105. Соколов А.В. и Степанюк О.М. «Методы информационной защиты объектов и компьютерных сетей», *СПб : АСТ*, с. 269, 2000.

106. Стасюк А.И. и Корченко А.А. «Базовая модель параметров для построения систем выявления атак», *Захист інформації*, №2 (55), с. 47–51, 2012.

107. Стасюк А.И. і Корченко А.А. «Метод виявлення аномалій породжених кібератаками в комп'ютерних мережах», Захист інформації, т. 14, № 4, с. 127–132, 2012.

108. Створення творення державної інтегрованої інформаційної системи забезпечення управління рухомими об'єктами (зв'язок, навігація, спостереження) України - виклик XXI століття / В. В. Корнієнко [та ін.]., Наука та інновації. 2007. Т 3, № 1. С. 4 – 32.

109. Тарасенко В.П., Корченко О.Г. і Терейковський І.А. «Метод застосування продукційних правил для подання експертних знань в нейромережевих засобах розпізнавання мережевих атак на комп'ютерні системи», *Безпека інформації*, т. 19, № 3. с. 68–174, 2013.

110. Терейковський І.А. «Використання нейронної мережі з радіальними базисними функціями в задачах діагностики стану захищеності програмного забезпечення», *Науково-технічний збірник «Управління*

розвитком складних систем» Київського національного університету будівництва і архітектури, т. 3, с. 111–114, 2010.

111. Терейковський І.А. «Дослідження стійкості серверних технологій Java від атак на відмову», *Захист інформації*, № 4, с. 34–42, 2004.

112. Файнберг М.А. и Файнберг Е.А. «Управление в системах массового обслуживания», Зарубежная радиоэлектроника. №2, с. 45–48, 1997.

113. Фатеев В.А. и Бочков М.В. «Методика обнаружения несанкционированных процессов при выполнении прикладных программ, основанная на аппарате скрытых Марковских цепей», *Сб. докл. VI Междунар. конф. SCM'2003*, т. 1, с. 218–220, 2003.

114. Хорошко В.А. и Терейковский И.А. «Использование искусственных нейронных сетей в задачах распознавания атак на компьютерные системы», *Захист інформації*, № 3, с. 57–65, 2006.

115. Хорошко В.А. [и др.] «Методы и средства защиты информации», *Київ : Юниор*, 504 с, 2003.

116. Хоффман Л. «Современные методы защиты информации», *М. : Сов. радио*, 264 с, 1980.

117. Цыпкин Я.З. «Основы теории обучающихся систем», *М. : Наука*, 251 с, 1970.

118. Шведун В. «Організаційно-правове забезпечення державного регулювання інформаційної безпеки реклами», *Безпека інформації*, т. 21, № 2. с. 174–178, 2015.

119. Швец В. и Васянович В. «Локализация контрольных точек на характерных фрагментах изображения лица человека», *Безпека інформації*, т. 21, № 1, с. 35–39, 2015.

120. Швец В. «Необходимость защиты информации глобальных навигационных спутниковых систем GPS, ГЛОНАСС, ГАЛИЛЕО», *Безпека інформації*, т. 20, № 2, с. 185–192, 2014.

121. Шевченко А. і Кокотов О. «Метод оцінювання ризиків з урахуванням впливу механізмів захисту інформації на параметри безпроводових інформаційно-телекомунікаційних систем під час інформаційних операцій», *Безпека інформації*, т. 20, № 1, с. 7–11, 2014.

122. Шелехов І.В. і Бірюкова М.М. «Інформаційно-екстремальна оптимізація словника ознак», *Вісник СумДУ. Серія "Технічні науки"*, № 3, с. 46–54, 2012.

123. Штовба С.Д «Классификация обьектов на основе нечеткого логического вивода», *Exponenta Pro: Математика в приложених*, №1(5), с. 68−69, 2004.

124. Щеглов А.Ю. и Финков М.В. «Защита компьютерной информации от несанкционированного доступа», *СПб. : Наука и техника*, 384 с. 2004.

125. Юдін О.К. «Інформаційна безпека», *Нормативно-правове забезпечення: підручник. Київ : НАУ*, 640 с. 2011.

126. Юдін О.К. і Бучик С.С. «Правові аспекти формування системи державних інформаційних ресурсів», *Безпека інформації*, т. 20 (1). с. 76–82, 2014.

127. Юдін О.К., Бучик С.С., Чунарьова А.В. і Варченко О.І. «Методологія побудови класифікатора загроз державним інформаційним ресурсам», *Наукоємні технології*, № 2, с. 200–210, 2014.

128. Яремчук Ю.Е., Шиян А.А., Заступ И.В. и Бекетова Г.С. «Структурно – функциональные модели для взаимодействия субъектов в системах менеджмента информационной безопасности», *Тезисы докладов участников II международной научно-практической конференции «Актуальные вопросы обеспечения кибербезопасности и защиты информации»*, с. 183–186, 2016.

129. Ярочкин В.И. «Технические каналы утечки информации», *М. : ИПКИП*, 112 с. 1994.

130. Ярочкин В.И. «Безопасность информационных систем», *М. : Ось-89*, 320 с. 1996.

131. 2015 Attacks Statistics URL: http://www.hackmageddon.com/2016 /01/11/ 2015 – cyber-attacks-statistics/

132. Ali M. Ameer, Karmakar G. C. and Dooley L.S. «Review on Fuzzy Clustering Algorithms», *IETECH Journal of Advanced Computations*, Vol. 2 (3), p. 169–181, 2008.

133. Al–Jarrah O. and Arafat A. «Network Intrusion Detection System using attack behavior classification», *Information and Communication Systems (ICICS) 5th International Conference*, p. 1–6, 2014.

134. Ameziane E., Hassani A., El Kalam A., Bouhoula A., Abassi R. and Ait Ouahman A. «Integrity–OrBAC: a new model to preserve Critical Infrastructures integrity», *International Journal of Information Security*. 14 (4), p. 367–385, 2014.

135. Atighetchi M., Pal P.P., Jones C.C. and Rubel P. «Building Auto–Adaptive Distributed Applications: The QuO–APOD Experience», *Proceedings of 3rd International Workshop Distributed Auto–adaptive and Reconfigurable Systems (DARES).* p.74–84, 2003.

136. Atighetchi M., Pal P., Webber F., Schantz R., Jones C. and Loyal J. "Adaptive Cyberdefense for Survival and Intrusion Tolerance", *Internet Computing*. *Vol. 8*, No 6. p.25–33, 2004.

137. Callegari C., Gazzarrini L., Giordano S., Pagano M. and Pepe T. «Improving PCA-based anomaly detection by using multiple time scale analysis and Kullback–Leibler divergence», *International Journal of Communication Systems.* Vol. 27, № 10, p. 1731–1751, 2014.

138. CERT/CC Statistics 2008–2015. URL: http://www.cert.org/ tats/cert_ tats.html.

139. Chapman C. and Ward S. «Project Risk Management: processes, techniques and insights», Chichester, John Wiley, Vol. 1210, p. 1780–1791, 2003.

140. Chi S.D., Park J.S. and Jung K.C. «Network Security Modeling and Cyber At–tack Simulation Methodology», *LNCS*, Vol. 2119, p. 231–242, 2001.

141. Chinh H. N., Hanh T. and Dinh N.T. «Fast detection of DDOS attacks using non-adaptive group testing», *International Journal of Network Security & Its Applications (IJNSA)*, Vol.5, № 5, P. 63–71, *2013.*

142. Chung M., Mukherjee B., Olsson R.A. and Puketza N. "Simulating Concurrent Intrusions for Testing Intrusion Detection Systems", *Proc. of the 18th NISSC*, 1995.

143. Creating trust in the digital world EY's Global Information Security Survey 2015 URL: http://www.ey.com/Publication /vwLUAssets /ey-global-information-security-survey - 2015 /$FILE/ey-global-information-security-survey-2015.pdf

144. Dawkins J., Campbell C. and Hale J. "Modeling network attacks: Extending the attack tree paradigm,Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection". *Johns Hopkins University*, 600 p. 2002.

145. Goldman R.P. «A Stochastic Model for Intrusions», *LNCS*, Vol. 2516, p. 290–295, 2002.

146. Gorodetsk V. and Kotenko I. «Attacks against Computer Network: Formal Grammar–based Framework and Simulation Tool», *RAID 2000, LNCS*, Vol. 2516, p. 349–350, 2002.

147. Govindarajan M. and Chandrasekaran R.M. «Intrusion Detection Using an Ensemble of Classification Methods», *World Congress on Engineering and Computer Science,* Vol. 1, p. 459–464, 2012.

148. Guan Y., Ghorbani A.A. and Belacel N. «Y-means: a clustering method for intrusion detection», *In Canadian Conference on Electrical and Computer Engineering*, Vol. 2, p. 1083–1086, 2003.

149. Guitton C. and Korzak E. "The Sophistication Criterion for Attribution", *The RUSI Journa,.* Vol.158, Iss. 4, p. 62–68, 2013.

150. Gyanchandani M., Rana J.L. and Yadav R.N. "Taxonomy of anomaly based intrusion detection system: a review", *International Journal of Scientific and Research Publications*, Vol. 2, Iss. 12, p. 1–13, 2012.

151. Halkidi M., Batistakis Y. and Vazirgiannis M. «On Clustering Validation Techniques», *Journal of Intelligent Information Systems*, Vol. 17, Iss. 2–3, p. 107–145, 2001.

152. Harel Statecharts D. «A Visual Formalism for Complex Systems», *Science of Computer Programming*, N 8, p. 231–274, 1987.

153. Hariri S., Qu G. and Dharmagadda T. «Impact Analysis of Faults and Attacks in Large–Scale Networks», *IEEE Security & Privacy*, p. 456–459, 2003.

154. Hatley D.J. and Pirbhai I.A. «Strategies for Real–Time System Specification», *Dorset House Publishing Co. Inc., NY*, 930 p. 1988.

155. Heckerman D. «A tutorial on learning with bayesian networks», *Innovations in Bayesian Networks,* Vol. 156, P. 33–82, 2008.

156. Iglun K., Kemmerer R. and Porras P. «A State Transition Analysis: A Rule–Based Intrusion Detection System», *IEEE Transactions on Software Engineering*, N 21(3), P. 422–434, 1995.

157. Ishida Y. «Immunity–Based Systems A Design Perspective». *Springer Verlag*, 192 p, 2004.

158. ISO/IEC IS 27001:2005 Information technology. Security techniques. Information security management systems. Requirements.

159. Ismail M.N., Aborujilah A., Musa S. and Shahzad A. «Detecting flooding based DoS attack in cloud computing environment using covariance matrix approach», *ICUIMC '13 Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, P. 1–7, 2013.

160. Jha S., Sheyner O. and Wing J. «Minimization and reliability analysis of attack graphs», *Technical Report CMU–CS–02–109. Carnegie Mellon University*, 79 p. 2002.

161. Jyothsna V. and Prasad Rama V.V. «A review of anomaly based intrusion detection systems», *International Journal of Computer Applications*, Vol. 28, No. 7, p. 26–35, 2011.

162. Kabiri P. and Ghorbani A. «A research on intrusion detection and response: a survey», *International Journal of Network Security*, Vol. 1, N 2, p. 84–102, 2005.

163. Keromytis A.D., Parekh J., Gross P.N., Kaiser G., Misra V., Nieh J., Rubensteiny D. and Stolfo S. «A Holistic Approach to Service Survivability», *Proceedings of ACM Workshop on Survivable and Self–Regenerative Systems*, p.11–22, 2003.

164. Khan L., Awad M. and Thuraisingham B. «A new intrusion detection system using support vector machines and hierarchical clustering», *The International Journal on Very Large Data Bases*, Vol. 16, Iss. 4, p. 507–521, 2007.

165. Khattab S., Gobriel S., Melhem R. and Moss´e D. "Live Baiting for Service-Level DoS Attackers", *INFOCOM 2008. The 27th Conference on Computer Communications IEEE*, P. 1–9, 2008.

166. Komar M., Golovko V., Sachenko A. and Bezobrazov S. «Development of Neural Network Immune Detectors for Computer Attacks Recognition and Classification», *IEEE 7th Intern. Conf. on Intelligent Data Acquisition and Advanced Computing Systems*, Vol. 2, p. 665–668, 2013.

167. Korobiichuk, I., Hryshchuk, R., Mamarev, V., Okhrimchuk, V., & Kachniarz, M. «Cyberattack Classificator Verification. In International Conference on Diagnostics of Processes and Systems», *Springer, Cham*, p. 402-411, 2017.

168. Lakhno, Y. Tkach, T. Petrenko, S. Zaitsev and V. Bazylevych, "Development of adaptive expert system of information security using a procedure of clustering the attributes of anomalies and cyber attacks", *Eastern-European Journal of Enterprise Technologies,* no. 6/9 (84), pp. 32–44, 2016.

169. V. Lakhno, S. Zaitsev, Y. Tkach, T. Petrenko, "Adaptive expert systems development for cyber attacks recognition in information educational systems on the basis of signs' clustering", *Advances in Intelligent Systems and Computing*, v. 754. pp. 673–682, 2018.

170. Lu W. and Traore I. «Detecting new forms of network intrusion using genetic programming», *Computational intelligence*, Vol. 20, No 3, p. 475–494, 2004.

171. Lye K. and Wing J. «Game Strategies in Network Security,International Journal of Information Security», N 4, p. 340–349, 2005.

172. McNab C. «Network Security Assessment», *O'Reilly Media, Inc.*, 450 p. 2004.

173. Mirkovic J., Dietrich S., Dittrich D. and Reiher P. «Internet Denial of Service: Attack and Defense Mechanisms». *Prentice Hall PTR*, 400 p. 2004.

174. MITRE Research Program URL: http://www.mitre.org

175. Moitra S.D. and Konda S.L. «A Simulation Model for Managing Survivability of Net–worked Information Systems», *Technical Report CMU/SEI–2000–TR–020*, 189 p. 2000.

176. Mukkamala S., Sung A.H., Abraham A. and Ramos V. «Intrusion detection systems using adaptive regression splines», *Sixth International Conference on Enterprise Information Systems*, Part 3, p, 211–218, 2013.

177. Negoita M., Neagu D. and Palade V. "Computational Intelligence Engineering of Hybrid Systems", *Springer Verlag*, 213 p. 2005.

178. Nessus Network Auditing. Renaud Deraison. Syngress Publishing, Inc., 2004.

179. Omar S., Ngadi A. and Jebur H.H. «Machine learning techniques for anomaly detection: an overview», *International Journal of Computer Applications*, Vol. 79, No. 2, p. 33–41, 2013.

180. Ortalo R., Dewarte Y. and Kaaniche M. «Experimenting with quantitative evaluation tools for monitoring operational security», *IEEE Trans. on Software Engineering*, N 25 (5), p. 570–582, 1999.

181. OSVDB: The Open Source Vulnerability Database URL: http://www.osvdb.org/

182. Pawar S.N. «Intrusion detection in computer network using genetic algorithm approach: a survey», *International Journal of Advances in Engineering Technology*. Vol. 6, Iss. 2, p. 730–736, 2013.

183. Peddabachigari S., Abraham A., Grosan C. and Thomas J. «Modeling intrusion detection system using hybrid intelligent systems», *Joural of Network and Computer Applications*, Vol. 30, Iss. 1, p, 114–132, 2007.

184. Piszcz A., Orlans N. and Eyler–Walker Z. «Moore D. Engineering Issues for an Adaptive Defense Network», *MITRE Technical Report*, 70 p, 2001.

185. Powers S.T. and He J. «A hybrid artificial immune system and self organising map for network intrusion detection», *Information Sciences*, Vol. 178, Iss. 5, p. 3024–3042, 2008.

186. Protect Yourself Against Social Engineering Attacks. URL: http://www.dhs.gov

187. Raiyn J. «A survey of Cyber Attack Detection Strategies», *International Journal of Security and Its Applications*, Vol. 8, No.1, p. 247–256, 2014.

188. Ranjan R. and Sahoo G. «A new clustering approach for anomaly intrusion detection», *International Journal of Data Mining Knowledge Management Process*, Vol. 4, No. 2, p. 29–38, 2014.

189. Results of internet SSL usage published by SSL Labs URL: http://webappsec.org

190. Riadi I., Istiyanto J.E., Ashari A. and Subanar N. «Log Analysis Techniques using Clustering in Network Forensics», *International Journal of Computer Science and Information Security*, Vol. 10, No.7, p. 740–749, 2013

191. Rohse M. and Bugtraq C. «Vulnerability naming schemes and description languages». *SANS GSEC PRACTICAL*, 2003.

192. Selim S., Hashem M. and Nazmy T.M. «Detection using multi–stage neural network», *International Journal of Computer Science and Information Security*, Vol. 8, No. 4, p. 14–20, 2010.

193. Stasiuk A. I., Hryshchuk R. V. and Goncharova L. L. "A Mathematical Cybersecurity Model of a Computer Network for the Control of Power Supply of Traction Substations." *Cybernetics and Systems Analysis 53.3*, (2017): 476-484.

194. Sumeet Dua. and Xian Du. «Data Mining and Machine Learning in Cybersecurity». *New York: CRC Press*, 256 p. 2011.

195. Stewart A.J. «Distributed Metastasis: A Computer Network Penetration Methodology», *Phrack Magazine*, № 9 (55), p. 449–451, 1999.

196. Takagi H. and Wu D. «A Multiserver queue with semi – Markovian batch arrivals with application to the MPEG frame sequence, Internet Performance and Control of Network System III», *Proceeding of SPIE*, Vol. 4865, p. 178–189, 2002.

197. Thai M.T., Xuan Y., Shin I. and Znati T. «On Detection of Malicious Users Using Group Testing Techniques», *Distributed Computing Systems*, p. 206 – 213, 2008.

198. The Web Hacking Incidents Database 2008: Annual Report. URL: http://www.breach.com/confirmation/2008WHID.html

199. Toosi A.N. and Kahani M.A «New Approach to Intrusion Detection Based on an Evolutionary Soft Computing Model Using Neuro-Fuzzy Classifiers», *Computer Communications*, Vol. 30, Iss. 10, p. 2201–2212, 2007.

200. Tsai C.F. , Hsub Y.F., Linc C.–Y. and Lin W.–Y. «Intrusion detection by machine learning: a review», *Expert Systems with Applications*, Vol. 36, Iss. 10, p. 11994–12000, 2009.

201. Unsupervised adaptive filtering. V. 1, 2. Edited by S. Haykin. New York: John Willey & Sons, Inc, 2000. 1402 p.

202. Urchin V., Peng T., Leckie C. and Ramamohanarao K. «Survey of Network– Based Defense Mechanisms Countering the DoS and DDoS Problems», *ACM Computing Surveys*, Vol. 39, № 1, P. 31–42, 2007.

203. Urgen M.W. and Stolfo S. “Ensemble–based adaptive intrusion detection”, *SIAM International Conference on Data Mining*, 2002.

204. Vasiliu E.V. “Non-coherent attack on the ping-pong protocol with completely entangled pairs of qutrits”, *Quantum Information Processing*, V. 10, N 2, p. 189–202, 2011.

205. Vinchurkar D. and Reshamwala M. «A review of intrusion detection system using neural network and machine learning technique», *International Journal of Engineering Science and Innovative Technology*, Vol. 1, № 2, p. 54–63, 2012.

206. Walk T. «Cyber-attack protection for pipeline SCADA systems», *Pipelines International digest*, P. 5–8, 2012.

207. Wu S. and Banzhaf W. “The use of computational intelligence in intrusion detection systems: a review”, *Applied Soft Computing*, Vol. 10, Is. 1, p. 1–35, 2010.

208. Xiang Y., Zhou W. and Chowdhury M. “ ASurvey of Active and Passive Defence Mechanisms against DDoS Attacks”, *Technical Report, TR C04/02, School of Information Technology, Deakin University*, 2004.

209. Xiang Y., Li K. and Zhou W. “Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics”, *IEEE Transactions on Information Forensics and Security*, Vol. 6, № 2, P. 426 – 437, 2011.

210. Zhou Y.P. “Hybrid Model Based on Artificial Immune System and PCA Neural Networks for Intrusion Detection”, *Asia–Pacific Conference on Information Processin*g, Vol. 1, P. 21–24, 2009.

211. Zou C.C., Duffield N., Towsley D. and Gong W. «Adaptive Defense against Various Network Attacks», *IEEE Journal on Selected Areas in Communications: High–Speed Network Security*, Vol. 24, №.10, p. 44 –51, 2006.

Додаток А. Відомості щодо впровадження результатів роботи

Україна

ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ «ІНФОРМАЦІЙНА БЕЗПЕКА»

14030, м. Чернігів, вул. Захисників України, 25, тел. (095) 285-01-44

№ 134 від 16.11.2018 р.

АКТ
про впровадження результатів дисертаційного дослідження
Петренка Тараса Анатолійовича

Цим Актом засвідчується, що наступні наукові та практичні результати досліджень, які розроблені в дисертаційній роботі на тему «Методи і моделі експертних систем розпізнавання кібератак на основі кластеризації реалізацій ознак» Петренка Тараса Анатолійовича, використовуються у ТОВ «Інформаційна безпека»:

1. Моделі та методи навчання експертної системи розпізнавання кіберзагроз з можливістю нечіткої кластеризації реалізацій ознак аномалій, кіберзагроз або кібератак, та корекцією вирішальних правил. Зазначені методи і моделі дозволяють скоротити час навчання системи кіберзахисту та підвищити ефективнісньо розпізнавання кібератак.
2. Експертна система «Аналізатор загроз» використовувалася у ТОВ «Інформаційна безпека» в період 2017-2018 р. для діагностування захищеності інформаційних систем та системи електронного документообігу.
3. Встановлено, що запропонований алгоритм інформаційно-екстремального навчання ЕС «Аналізатор загроз» є найбільш ефективним для 3 (трьох) кластерів в завданнях розбиття простору ознак аномалій та кіберзагроз. При цьому, в режимі тестового навчання ЕС достатня кількість кроків для безпомилкового визначення класів аномалій, кіберзагроз або кібератак склала 2500-3000.
4. Виявлення основних типів мережевих атак у разі використовування ЕС «Аналізатор загроз» відбувалося з ймовірністю 77–99 % при незначному рівні помилкових спрацювань.

Директор Зайцев С.В.

Головний фахівець із захисту інформації Усов Я.Ю.

Затверджую:
Проректор з навчальної роботи
Національного авіаційного
університету
_____ А. Гудманян
"3" вересня 2018 р.

АКТ

про впровадження результатів дисертаційного дослідження

Петренка Тараса Анатолійовича на здобуття наукового ступеня

кандидата технічних наук

Комісія у складі: завідувач кафедри безпеки інформаційних технологій (БІТ) Корченко О.Г. (голова), проф. кафедри БІТ Казмірчук С.В., проф. кафедри БІТ Іванченко Є.В. склали даний акт про те, що результати наукового дослідження Петренка Т.А. впроваджені в навчальний процес і використовувалися на кафедрі БІТ при викладанні дисципліни "Інтелектуалізовані системи інформаційної безпеки", що входить в навчальний план спеціальності "Кібербезпека".

№ п/п	Назва впроваджуваної моделі	Форма впровадження	Ефективність від впровадження
	1	2	3
1.	Моделі та алгоритм навчання адаптивної системи розпізнавання кіберзагроз з можливістю нечіткої кластеризації ознак аномалій, кіберзагроз або кібератак і корекцією вирішальних правил.	Лекція.	Систематизація навчального матеріалу і надання студентам знань про сучасні інетелектуалізовані нейромережеві методи протидії кібератакам.
2.	Дослідження експертної системи "Аналізатор загроз" для діагностування захищеності інформаційних систем.	Лабораторна Робота.	Ознайомлення з базовими підходами і навчання студентів практичним діям щодо аналізу загроз мережевим ресурсам інформаційних систем.

Голова комісії
завідувач кафедри БІТ, д.т.н., проф.,
лауреат Державної премії України
в галузі науки і техніки О.Корченко

Члени комісії:

проф., кафедри БІТ, д.т.н., доц. С.Казмірчук

проф., кафедри БІТ, к.т.н., доц. Є.Іванченко

ЗАТВЕРДЖУЮ
Проректор з науково-педагогічної роботи
Чернігівського національного
технологічного університету
Кальченко В.В.
"__" 09 2018 р.

Акт № 9 від 05.09.2018р
про впровадження в навчальний процес результатів дисертаційної роботи
за темою «Методи і моделі експертних систем розпізнавання кібератак на основі кластеризації реалізацій ознак»
старшого викладача кафедри «Кібербезпеки та математичного моделювання»
Петренка Тараса Анатолійовича

Ректорат Чернігівського національного технологічного університету підтверджує впровадження в навчальний процес результатів дисертаційної роботи на здобуття наукового ступеня кандидата технічних наук Петренка Тараса Анатолійовича за темою «Методи та моделі експертних систем розпізнавання кібератак на основі кластеризації реалізацій ознак» зі спеціальності 05.13.21 – системи захисту інформації.

Методи та моделі, розроблені автором впроваджено в навчальний процес при викладанні таких дисциплін: «Системи штучного інтелекту» для студентів спеціальності 125 – Кібербезпека та «Моделі та системи штучного інтелекту» для студентів спеціальності 123 – Комп'ютерна інженерія.

Дослідження розроблених автором методів та моделей експертних систем розпізнавання кібератак дозволяють студентам поглибити рівень знань і вмінь в галузі проектування, розробки та реалізації експертних систем призначених для розпізнавання аномалій, кіберзагроз та кібератак на інформаційні системи. Під час лабораторної роботи з експертною системою «Аналізатор загроз» студенти знайомляться з базовими підходами та практичними діями щодо аналізу загроз мережевим ресурсам інформаційних систем.

Завідувач кафедри кібербезпеки
та математичного моделювання _____ д.пед.н., доц.
Ю.М. Ткач

Завідувач кафедри інформаційних
і комп'ютерних систем _____ д.т.н., доц.
С.В. Зайцев

Ткач Ю.М, Зайцева С.В

Додаток Б. Лістинг ЕС «Analyzer of cyberthreats»

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, CommCtrl, Winsock, ImgList, ShellAPI, XPMan,
  jpeg, ExtCtrls, FileCtrl, AdvProgressBar, inifiles, registry,  IpHlpApi, IpTypes,  IpIfConst,
  Grids, AdvObj, BaseGrid, AdvGrid, AdvPanel, AdvGlowButton,
  AdvGridWorkbook, Buttons, rtflabel, TeeProcs, TeEngine, Chart, Series;
resourcestring
  RES_THREADCOUNT = 'Запущено потоков: %d';
  RES_COMPCOUNT = 'Найдено: %d';
  RES_ERR_RANGE = 'Недопустимый диапазон';
const
  TH32CS_SNAPPROCESS  = $00000002;
  // Константы состояний порта
  MIB_TCP_STATE_CLOSED     = 1;
  MIB_TCP_STATE_LISTEN     = 2;
  MIB_TCP_STATE_SYN_SENT   = 3;
  MIB_TCP_STATE_SYN_RCVD   = 4;
  MIB_TCP_STATE_ESTAB      = 5;
  MIB_TCP_STATE_FIN_WAIT1  = 6;
  MIB_TCP_STATE_FIN_WAIT2  = 7;
  MIB_TCP_STATE_CLOSE_WAIT = 8;
  MIB_TCP_STATE_CLOSING    = 9;
  MIB_TCP_STATE_LAST_ACK   = 10;
  MIB_TCP_STATE_TIME_WAIT  = 11;
  MIB_TCP_STATE_DELETE_TCB = 12;
  STR_START    =  'Начать сканирование';
  STR_STOP     =  'Остановить сканирование';
  STR_STARTED  =  '   Идет сканирование ...';
  STR_STOPPED  =  '   Сканирование завершено ...';
  STR_END      =  '   Завершение потока ...';
  STR_FIELD    =  '   Поле не выбрано ...';
  WSA_TYPE = $101;
    RES_UNKNOWN = 'Неизвестно';
    RES_IP      = 'IP адрес: ';
    RES_CMP     = 'Имя компьютера: ';
    RES_USR     = 'Имя пользователя: ';
    RES_DOM     = 'Домен: ';
    RES_SER     = 'Сервер домена: ';
    RES_COM     = 'Коментарий: ';
    RES_PROV    = 'Провайдер: ';
    RES_GRP     = 'Группы: ';
    RES_MAC     = 'MAC адресс: ';
    RES_SHARES  = 'Доступные ресурсы: ';
    RES_TIME    = 'Времени затрачено: ';
    RES_COM_NO  = 'Отсутствует';
 // Для работы с ARP (Address Resolution Protocol) таблицей
  IPHLPAPI = 'IPHLPAPI.DLL';
```

```
  MAX_ADAPTER_ADDRESS_LENGTH = 7;
type
     // Расширенные варианты данных структур
    PTMibTCPExRow = ^TMibTCPExRow;
    TMibTCPExRow = packed record
      dwState: DWORD;
      dwLocalAddr: DWORD;
      dwLocalPort: DWORD;
      dwRemoteAddr: DWORD;
      dwRemotePort: DWORD;
      dwProcessID: DWORD;
    end;
    PTMibTCPExTable = ^TMibTCPExTable;
    TMibTCPExTable = packed record
      dwNumEntries: DWORD;
      Table: array[0..0] of TMibTCPExRow;
    end;
    PTMibUdpExRow = ^TMibUdpExRow;
    TMibUdpExRow = packed record
      dwLocalAddr: DWORD;
      dwLocalPort: DWORD;
      dwProcessID: DWORD;
    end;
    PTMibUdpExTable = ^TMibUdpExTable;
    TMibUdpExTable = packed record
      dwNumEntries: DWORD;
      table: array [0..0] of TMibUdpExRow;
    end;
    // Структура для получения списка текущий процессов и их параметров
    TProcessEntry32 = packed record
      dwSize: DWORD;
      cntUsage: DWORD;
      th32ProcessID: DWORD;
      th32DefaultHeapID: DWORD;
      th32ModuleID: DWORD;
      cntThreads: DWORD;
      th32ParentProcessID: DWORD;
      pcPriClassBase: Longint;
      dwFlags: DWORD;
      szExeFile: array [0..MAX_PATH - 1] of WideChar;
    end;
  // Стандартная структура для получения TCP статистики
  PTMibTCPRow = ^TMibTCPRow;
  TMibTCPRow = packed record
    dwState: DWORD;
    dwLocalAddr: DWORD;
    dwLocalPort: DWORD;
    dwRemoteAddr: DWORD;
    dwRemotePort: DWORD;
  end;
```

```
// В данную структуру будет передаваться результат GetTcpTable
PTMibTCPTable = ^TMibTCPTable;
TMibTCPTable = packed record
  dwNumEntries: DWORD;
  Table: array[0..0] of TMibTCPRow;
end;
// Стандартная структура для получения UDP статистики
PTMibUdpRow = ^TMibUdpRow;
TMibUdpRow = packed record
  dwLocalAddr: DWORD;
  dwLocalPort: DWORD;
end;
// В данную структуру будет передаваться результат GetUDPTable
PTMibUdpTable = ^TMibUdpTable;
TMibUdpTable = packed record
  dwNumEntries: DWORD;
  table: array [0..0] of TMibUdpRow;
end;
// Структура для определения принадлежности пользователя к группам
PGroupUsersInfo0 = ^_GROUP_USERS_INFO_0;
_GROUP_USERS_INFO_0 = packed record
  grui0_name: LPWSTR;
end;
TGroupUsersInfo0 = _GROUP_USERS_INFO_0;
GROUP_USERS_INFO_0 = _GROUP_USERS_INFO_0;
 _WKSTA_USER_INFO_1 = record
  wkui1_username: LPWSTR;
  wkui1_logon_domain: LPWSTR;
  wkui1_oth_domains: LPWSTR;
  wkui1_logon_server: LPWSTR;
end;
WKSTA_USER_INFO_1 = _WKSTA_USER_INFO_1;
PWKSTA_USER_INFO_1 = ^_WKSTA_USER_INFO_1;
LPWKSTA_USER_INFO_1 = ^_WKSTA_USER_INFO_1;
// MAC
TMacAddress = array[0..MAX_ADAPTER_ADDRESS_LENGTH] of byte;
// Структура для единичного запроса
TMibIPNetRow = packed record
  dwIndex         : DWORD;
  dwPhysAddrLen   : DWORD;
  bPhysAddr       : TMACAddress;  // Здесь  лежит MAC!!!
  dwAddr          : DWORD;
  dwType          : DWORD;
end;
TMibIPNetRowArray = array [0..512] of TMibIPNetRow;
PTMibIPNetTable = ^TMibIPNetTable;
TMibIPNetTable = packed record
  dwNumEntries    : DWORD;
  Table: TMibIPNetRowArray;
end;
TDemoThread = class(TThread)
private
```

```
    TreeNetWrk: TTreeNode;
    TreeDomain: TTreeNode;
    TreeServer: TTreeNode;
    TreeShares: TTreeNode;
    Param_dwType: Byte;
    Param_dwDisplayType: Byte;
    Param_lpRemoteName: String;
    Param_lpIP: String;
  protected
    procedure Execute; override;
    procedure Scan(Res: TNetResource; Root: boolean);
    procedure AddElement;
    procedure Stop;
  end;
  TIPEdit = class
  private
    FHandle: THandle;
    FIP: Integer;
    FFont: Integer;
    function GetText: String;
    procedure SetText(const Value: String);
  public
    constructor Create(AOwner: TWinControl; Rect: TRect);
    destructor Destroy; override;
    property Text: String read GetText write SetText;
  end;
  TForm1 = class(TForm)
// Элементы интерфейса
    …..
//Функции
    function GetNameFromIP(const IP: String): String;
    function GetUsers(const CompName: String): String;
    function GetDomain(const CompName, Provider: String): String;
    function GetComment(CompName, Provider: String): String;
    function GetProvider(const CompName: String): String;
    function GetMacFromIP(const IP: String): String;
    function GetDomainServer(const DomainName: String): String;
    function GetGroups(DomainServer: String; UserName: String): String;
    function GetShares(const CompName: String): String;
//Процедуры
    ….
//Переменные
  private
    Thread: TDemoThread;  IPFrom, IPTo: TIPEdit; FThreadCount, FCompFound: Integer;
    IP, Font: Integer; IPst:String;
    function PortStateToStr(const State: DWORD): String;
    procedure SetThreadCount(const Value: Integer);
    procedure SetCompFound(const Value: Integer);
    procedure GetMemoryInfo;
    procedure GetCompInfo;
    procedure GetIPInfo;
    procedure GetAdapterInfo;
```

```
    procedure UpdateDisk;
  public
    property ThreadCount: Integer read FThreadCount write SetThreadCount;
    property CompFound: Integer read FCompFound write SetCompFound;
  end;
  LMSTR = LPWSTR;
  NET_API_STATUS = DWORD;
  PShareInfo1 = ^_SHARE_INFO_1;
  _SHARE_INFO_1 = record
   shi1_netname: LMSTR;
   shi1_type: DWORD;
   shi1_remark: LMSTR;
  end;
  TShareInfo1 = _SHARE_INFO_1;
  TScanThread = class(TThread)
  private
    FIP: Integer;
    FRes: TStringList;
    function GetCompName(const Addr: Integer): String;
    procedure Scan;
    procedure UpdateTree;
    procedure IncCount;
    procedure DecCount;
  protected
    procedure Execute; override;
  public
    property IP: Integer read FIP write FIP;
  end;
  // Объявим функции, так как их объявлений нет в Дельфи.
  // Здесь идет статическая загрузка библиотек, только потому,
  // что данные функции есть во всех системах, начиная с W95...
  {$EXTERNALSYM WNetGetResourceInformation}
  function WNetGetResourceInformation(lpNetResource: PNetResource;
    lpBuffer: Pointer; var lpcbBuffer: DWORD; lplpSystem: Pointer): DWORD; stdcall;
  {$EXTERNALSYM GetIpNetTable}
  function GetIpNetTable(pIpNetTable: PTMibIPNetTable;
    pdwSize: PULONG; bOrder: Boolean): DWORD; stdcall;
  function WNetGetResourceInformation; external mpr name 'WNetGetResourceInformationA';
  function GetIpNetTable; external IPHLPAPI name 'GetIpNetTable';
  function NetGetAnyDCName(servername: LPCWSTR;  domainname: LPCWSTR;
    bufptr: Pointer): Cardinal;
    stdcall; external 'netapi32.dll';
  function NetShareEnum(servername: LMSTR; level: DWORD; var bufptr: Pointer;
   prefmaxlen: DWORD; entriesread, totalentries,
   resume_handle: LPDWORD): NET_API_STATUS; stdcall; external 'Netapi32.dll';
  function NetApiBufferFree(Buffer: Pointer): NET_API_STATUS; stdcall; external
'Netapi32.dll';
  function NetWkstaUserEnum(ServerName: LPCWSTR;
                      Level: DWORD;
                      BufPtr: Pointer;
                      PrefMaxLen: DWORD;
                      EntriesRead: LPDWORD;
```

```
                    TotalEntries: LPDWORD;
                    ResumeHandle: LPDWORD): LongInt; stdcall; external 'netapi32.dll';
  function NetUserGetGroups(ServerName: LPCWSTR;
                    UserName: LPCWSTR;
                    level: DWORD;
                    bufptr: Pointer;
                    prefmaxlen: DWORD;
                    var entriesread: DWORD;
                    var totalentries: DWORD): LongInt; stdcall; external 'netapi32.dll';
  procedure InfComp;
  function GetTcpTable(pTCPTable: PTMibTCPTable; var pDWSize: DWORD;
    bOrder: BOOL): DWORD; stdcall; external 'IPHLPAPI.DLL';
  function GetUdpTable(pUDPTable: PTMibUDPTable; var pDWSize: DWORD;
    bOrder: BOOL): DWORD; stdcall; external 'IPHLPAPI.DLL';
    function AllocateAndGetTcpExTableFromStack(pTCPExTable: PTMibTCPExTable;
      bOrder: BOOL; heap: THandle; zero: DWORD; flags: DWORD): DWORD; stdcall;
      external 'IPHLPAPI.DLL';
    function AllocateAndGetUdpExTableFromStack(pUDPExTable: PTMibUDPExTable;
      bOrder: BOOL; heap: THandle; zero: DWORD; flags: DWORD): DWORD; stdcall;
      external 'IPHLPAPI.DLL';
    function CreateToolhelp32Snapshot(dwFlags, th32ProcessID: DWORD): THandle;
      stdcall; external 'KERNEL32.DLL';
    function Process32First(hSnapshot: THandle; var lppe: TProcessEntry32): BOOL;
      stdcall; external 'KERNEL32.DLL' name 'Process32FirstW';
    function Process32Next(hSnapshot: THandle; var lppe: TProcessEntry32): BOOL;
      stdcall; external 'KERNEL32.DLL' name 'Process32NextW';
type
  TRes = record
    S: Extended;  Q: Extended;  end;
const
  N = 10;
var
  Form1: TForm1;
  Y, O, Z, E_, F1, F2, Res_S, Res_Q, B: array[1..255] of Extended;
  F, C, E, OO: array[1..255] of array[1..255] of Extended;
  arOp:array [1..7,1..4]of string;
implementation
{$R *.dfm}
function RoundFloat(R: Extended; Decimals: Integer): Extended;
var
  Factor: Extended;
Begin   Factor := Int(Exp(Decimals * Ln(10)));  Result := Round(Factor * R) / Factor; end;
function M(j: Integer): TRes;
var
  i, i1, i2: Integer;
  S1, S2, Q1, Q2: Extended;
begin
  // вычисляем S
  S1 := 0;
  for i := 1 to N do
    S1 := S1 + O[i]*F[i,j];
  for i1 := 1 to N do
```

```
    for i2 := 1 to N do
      if (i2 <> i1) then
        S1 := S1 + OO[i1,i2]*F[i1,j]*F[i2,j];
  S1 := abs(S1 - Y[j]);
  //ShowMessage(Floattostr(S1));
  S2 := 0;
  for i := 1 to N do
    S2 := S2 + O[i]*C[i,j];
  for i1 := 1 to N do
    for i2 := 1 to N do
      if (i2 <> i1) then
        S2 := S2 + OO[i1,i2]*(F[i1,j]*C[i2,j] + F[i2,j]*C[i1,j]);
  M.S := S1/S2;
  Q1 := 0;
  for i := 1 to N do
    Q1 := Q1 + O[i]*F[i,j];
  for i1 := 1 to N do
    for i2 := 1 to N do
      if (i2 <> i1) then
        Q1 := Q1 + OO[i1,i2]*F[i1,j]*F[i2,j];
  Q1 := abs(Y[j] - Q1);
  Q2 := 0;
  for i := 1 to N do
    Q2 := Q2 + O[i]*E[i,j];
  for i1 := 1 to N do
    for i2 := 1 to N do
      if (i2 <> i1) then
        Q2 := Q2 + OO[i1,i2]*(F[i1,j]*E[i2,j] + F[i2,j]*E[i1,j]);
  M.Q := Q1/Q2;
end;

function Replace(Str, X, Y: string): string;
{Str - строка, в которой будет производиться замена.
 X - подстрока, которая должна быть заменена.
 Y - подстрока, на которую будет произведена заменена}
var
  buf1, buf2, buffer: string;
  i: Integer;
begin
  buf1 := '';
  buf2 := Str;
  Buffer := Str;
  while Pos(X, buf2) > 0 do
  begin
    buf2 := Copy(buf2, Pos(X, buf2), (Length(buf2) - Pos(X, buf2)) + 1);
    buf1 := Copy(Buffer, 1, Length(Buffer) - Length(buf2)) + Y;
    Delete(buf2, Pos(X, buf2), Length(X));
    Buffer := buf1 + buf2;
  end;
  Replace := Buffer;
end;
procedure TForm1.SetCompFound(const Value: Integer);
```

```
begin
  FCompFound := Value;
  StatusBar1.Panels.Items[1].Text := Format(RES_COMPCOUNT, [Value]);
  Application.ProcessMessages;
end;
procedure TForm1.SetThreadCount(const Value: Integer);
begin
  if Value < FThreadCount then
    ProgressBar.Position := ProgressBar.Max - Value;
  FThreadCount := Value;
  StatusBar1.Panels.Items[0].Text := Format(RES_THREADCOUNT, [Value]);
  if Value = 0 then
  begin
    ProgressBar.Position := 0;
    btnStart.Enabled := True;
  end;
  Application.ProcessMessages;
end;
procedure ScanLocalNW();
begin
  with form1 do begin
    Tag := Tag + 1;
    if (Tag mod 2) = 1 then begin
      TreeView1.Items.Clear;
      StatusBar1.Panels[1].Text := STR_STARTED;
      Thread := TDemoThread.Create(False);
    end
    else  begin
      StatusBar1.Panels[1].Text := STR_END;
      Thread.Terminate;
    end;   end;   end;
function GetIPAddress(NetworkName: String): String;
var
 Error: DWORD;
 HostEntry: PHostEnt;
 Data: WSAData;
 Address: In_Addr;
begin
  Delete(NetworkName, 1, 2);
  Error:=WSAStartup(MakeWord(1, 1), Data);
  if Error = 0 then
  begin
    HostEntry:=gethostbyname(PChar(NetworkName));
    Error:=GetLastError;
    if Error = 0 then
    begin
      Address:=PInAddr(HostEntry^.h_addr_list^)^;
      Result:=inet_ntoa(Address);
    End  else
     Result:='Unknown';
  End   else
    Result:='Error';
```

```
  WSACleanup;
end;
{ TDemoThread }
procedure TDemoThread.Execute;
var
  R:TNetResource;
begin
  inherited;
  Priority := tpIdle;
  FreeOnTerminate := True;
  Resume;
  Scan(R, True);
  TreeDomain := nil;
  TreeServer := nil;
  Synchronize(Stop);
end;
procedure TDemoThread.Scan(Res: TNetResource; Root: boolean);
var
 hEnum: Cardinal;
 nrResource: array[0..512] of TNetResource;
 dwSize: DWORD;
 numEntries: DWORD;
 I: DWORD;
 dwResult: DWORD;
begin
  if Root then
    dwResult := WNetOpenEnum(RESOURCE_GLOBALNET, RESOURCETYPE_ANY,
      0, nil, hEnum)
  else
    dwResult := WNetOpenEnum(RESOURCE_GLOBALNET, RESOURCETYPE_ANY,
      0, @Res, hEnum);
  if dwResult = NO_ERROR then
  begin
    dwSize := SizeOf(nrResource);
    numEntries := DWORD(-1);                          // ERROR_NO_MORE_ITEMS
    if WNetEnumResource(hEnum, numEntries, @nrResource, dwSize) = NO_ERROR then
    begin       for i := 0 to numEntries - 1 do
      begin         if Terminated then Break;
        with nrResource[i] do
        begin
          Param_dwType := dwType;
          Param_dwDisplayType := dwDisplayType;
          Param_lpRemoteName := lpRemoteName;
          if Param_dwDisplayType = RESOURCEDISPLAYTYPE_SERVER then
            Param_lpIP := GetIPAddress(Param_lpRemoteName);
        end;
        if Assigned(nrResource[i].lpRemoteName) then
          Synchronize(AddElement);
        Scan(nrResource[i], false);
      end;
    WNetCloseEnum(hEnum);
    end;   end; end;
```

```
procedure TDemoThread.AddElement;
begin
  Application.ProcessMessages;
  case Param_dwDisplayType of
    RESOURCEDISPLAYTYPE_NETWORK:
    begin
      TreeNetWrk := Form1.TreeView1.Items.Add(nil, Param_lpRemoteName);
      TreeNetWrk.StateIndex := 1;
    end;
    RESOURCEDISPLAYTYPE_DOMAIN:
    begin
      TreeDomain := Form1.TreeView1.Items.AddChild(TreeNetWrk, Param_lpRemoteName);
      TreeDomain.StateIndex := 2;
    end;
    RESOURCEDISPLAYTYPE_SERVER:
    begin
      TreeServer := Form1.TreeView1.Items.AddChild(TreeDomain, Param_lpRemoteName + '
IP: ' + Param_lpIP);
      TreeServer.StateIndex := 3;
    end;
    RESOURCEDISPLAYTYPE_SHARE:
    begin
      TreeShares := Form1.TreeView1.Items.AddChild(TreeServer, Param_lpRemoteName);
      TreeShares.StateIndex := 3 + Param_dwType;
    end;   end;  end;
procedure TDemoThread.Stop;
begin
  Form1.StatusBar1.Panels[1].Text := STR_STOPPED;
  Form1.Tag := 0;
end;
{ TForm1 }
procedure ReadFile();
var
   i,j,v_pos1,v_pos2:Integer;
   str1,str2:string;
   f : TStringList;
begin
  f := TStringList.Create();
  f.LoadFromFile('tab.dbf');
  for i := 0 to f.Count-1 do begin
    v_pos1:=1;
    str1:=f[i];
    for j:=0 to 3 do begin
      v_pos2:=pos(' ',str1);
      str2:=copy(str1,1,v_pos2-1);
      form1.AdvStringGrid2.Cells[j+1,i+1]:=str2;
      delete(str1,1,v_pos2);
      v_pos1:=v_pos2;
    end;   end;
  f.Free;
  f := TStringList.Create();
  f.LoadFromFile('tab2.dbf');
```

```
  v_pos1:=1;
  str1:=f[0];
  for j:=0 to 3 do begin
     v_pos2:=pos(' ',str1);
     str2:=copy(str1,1,v_pos2-1);
     form1.AdvStringGrid1.Cells[1,j]:=str2;
     delete(str1,1,v_pos2);
     v_pos1:=v_pos2;
  end;
  f.Free;
  f := TStringList.Create();
  f.LoadFromFile('tab3.dbf');
  v_pos1:=1;
  str1:=f[0];
  for j:=0 to 6 do begin
     v_pos2:=pos(' ',str1);
     str2:=copy(str1,1,v_pos2-1);
     form1.AdvStringGrid3.Cells[1,j]:=str2;
     delete(str1,1,v_pos2);
     v_pos1:=v_pos2;
  end;
  f.Free;
end;
procedure WriteFile();
var
  f: textFile; //объявляем переменную
  str: string; //буфер для хранения прочитанных данных
  i,j:Integer;
begin
   AssignFile(f,'tab.dbf');
   Rewrite(f);
   for i := 0 to 6 do begin      str:='';
     for j:=0 to 3 do str:=str+form1.AdvStringGrid2.Cells[j+1,i+1]+' ';
     Writeln(f, str);
   end;
   Closefile(f);
   AssignFile(f,'tab2.dbf');
   Rewrite(f);
   str:='';
   for j:=0 to 3 do str:=str+form1.AdvStringGrid1.Cells[1,j]+' ';
   Writeln(f, str);
   Closefile(f);
   AssignFile(f,'tab3.dbf');
   Rewrite(f);
   str:='';
   for j:=0 to 6 do str:=str+form1.AdvStringGrid3.Cells[1,j]+' ';
   Writeln(f, str);
   Closefile(f);
end;
procedure TForm1.FormCreate(Sender: TObject);
var
  i,j:byte;
```

```
begin
  application.Title:='Анализатор уязвимости';
  Application.HintHidePause:=40000;
  Tag := 0;
  IPFrom := TIPEdit.Create(gbAddrRange, Rect(32, 16, 121, 21));
  IPFrom.Text := '192.168.1.4';
  IPTo := TIPEdit.Create(gbAddrRange, Rect(32, 40, 121, 21));
  IPTo.Text := '192.168.1.10';
   // Зададим первоначальный IP адрес (это адрес моей машины)
  IP := MAKEIPADDRESS(192, 168, 1, 4);
  ScanLocalNW;
  AdvStringGrid2.Cells[0,1]:='1. Определение информации подлежащей защите';
  AdvStringGrid2.Cells[0,2]:='2. Выявление угроз и каналов утечки информации';
  AdvStringGrid2.Cells[0,3]:='3. Проведение оценки уязвимости и рисков';
  AdvStringGrid2.Cells[0,4]:='4. Определение требований к СЗИ';
  AdvStringGrid2.Cells[0,5]:='5. Осуществление выбора средств защиты';
  AdvStringGrid2.Cells[0,6]:='6. Внедрение и использование выбранных мер и средств';
  AdvStringGrid2.Cells[0,7]:='7. Контроль целостности и управление защитой';
  AdvStringGrid3.Cells[0,0]:='1. Определение информации подлежащей защите';
  AdvStringGrid3.Cells[0,1]:='2. Выявление угроз и каналов утечки информации';
  AdvStringGrid3.Cells[0,2]:='3. Проведение оценки уязвимости и рисков';
  AdvStringGrid3.Cells[0,3]:='4. Определение требований к СЗИ';
  AdvStringGrid3.Cells[0,4]:='5. Осуществление выбора средств защиты';
  AdvStringGrid3.Cells[0,5]:='6. Внедрение и использование выбранных мер и средств';
  AdvStringGrid3.Cells[0,6]:='7. Контроль целостности и управление защитой';
  AdvStringGrid2.Cells[1,0]:='Уровень   1';
  AdvStringGrid2.Cells[2,0]:='Уровень   2';
  AdvStringGrid2.Cells[3,0]:='Уровень   3';
  AdvStringGrid2.Cells[4,0]:='Уровень   4';
  AdvStringGrid4.Cells[0,0]:='Уровень   1';
  AdvStringGrid4.Cells[1,0]:='Уровень   2';
  AdvStringGrid4.Cells[2,0]:='Уровень   3';
  AdvStringGrid4.Cells[3,0]:='Уровень   4';
  AdvStringGrid1.Cells[0,0]:='Уровень   1';
  AdvStringGrid1.Cells[0,1]:='Уровень   2';
  AdvStringGrid1.Cells[0,2]:='Уровень   3';
  AdvStringGrid1.Cells[0,3]:='Уровень   4';
  AdvStringGrid8.Cells[0,0] := 'Диапазон значений';
  AdvStringGrid8.Cells[0,1] := 'в %';
  AdvStringGrid8.Cells[0,2] := 'нет(0) да(1)';
  AdvStringGrid8.Cells[0,3] := 'Мбит';
  AdvStringGrid8.Cells[0,4] := 'нет(0) да(1)';
  AdvStringGrid8.Cells[0,5] := 'Мбит';
  AdvStringGrid8.Cells[0,6] := 'количество ...';
  AdvStringGrid8.Cells[0,7] := 'нет(0) да(1)';
  AdvStringGrid8.Cells[0,8] := 'в %';
  AdvStringGrid8.Cells[0,9] := 'нет(0) да(1)';
  AdvStringGrid8.Cells[0,10] := 'в бит/с';
  AdvStringGrid9.Cells[0,0] := 'Результат вычислений';
  AdvStringGrid9.Cells[1,0] := 'S';
  AdvStringGrid9.Cells[2,0] := 'Q';
  AdvStringGrid7.Cells[0,0] := 'Критерии';
```

```
AdvStringGrid7.Cells[0,1] := 'Избыточный трафик';
AdvStringGrid7.Cells[0,2] := 'Наличие подозрительных процессов';
AdvStringGrid7.Cells[0,3] := 'Большие пакеты или фрагменты';
AdvStringGrid7.Cells[0,4] := 'Несоответствие IP адресов в ICMP-пакетах';
AdvStringGrid7.Cells[0,5] := 'Размер ICMP-пакетов';
AdvStringGrid7.Cells[0,6] := 'Открытые порты';
AdvStringGrid7.Cells[0,7] := 'Включенные опции в TCP/UDP-пакетах';
AdvStringGrid7.Cells[0,8] := 'Большое количество pingзапросов/ответов';
AdvStringGrid7.Cells[0,9] := 'Отсутствие ping-a';
AdvStringGrid7.Cells[0,10] := 'Замедление скорости соединения';
AdvStringGrid9.Cells[0,0] := 'По критериям';
AdvStringGrid9.Cells[0,1] := 'По  трафик';
AdvStringGrid9.Cells[0,2] := 'По наличию подозрительных процессов';
AdvStringGrid9.Cells[0,3] := 'По пакетам или фрагментам';
AdvStringGrid9.Cells[0,4] := 'По несоответствию IP адресов в ICMP-пакетах';
AdvStringGrid9.Cells[0,5] := 'По размеру ICMP-пакетов';
AdvStringGrid9.Cells[0,6] := 'По открытым портам';
AdvStringGrid9.Cells[0,7] := 'По включеннию опции в TCP/UDP-пакетах';
AdvStringGrid9.Cells[0,8] := 'По ping запросам/ответам';
AdvStringGrid9.Cells[0,9] := 'По отсутствию ping-a';
AdvStringGrid9.Cells[0,10] := 'По замедлению скорости соединения';
AdvStringGrid7.Cells[1,0] := 'Оценка';
AdvStringGrid7.Cells[1,1] := '5';
AdvStringGrid7.Cells[1,2] := '3';
AdvStringGrid7.Cells[1,3] := '7';
AdvStringGrid7.Cells[1,4] := '4';
AdvStringGrid7.Cells[1,5] := '8';
AdvStringGrid7.Cells[1,6] := '4';
AdvStringGrid7.Cells[1,7] := '5';
AdvStringGrid7.Cells[1,8] := '6';
AdvStringGrid7.Cells[1,9] := '9';
AdvStringGrid7.Cells[1,10] := '3';
for i := 1 to N do
  AdvStringGrid9.Cells[0,j] := Floattostr(i);
for i := 1 to N do
  AdvStringGrid8.Cells[i,0] := Floattostr(i);
for i := 1 to N do begin
  F[i,1] := i*10;
  F[i,2] := 1;
  F[i,3] := 96;
  F[i,4] := 1;
  F[i,5] := i*10;
  F[i,6] := i;
  F[i,7] := 1;
  F[i,8] := i*10;
  F[i,9] := 1;
  F[i,10] := i*10;
end;
for i:=1 to N do     for j:=1 to N do
    C[i,j]:= (i / j) / N + 0.1 ;   for i:=1 to N do
  for j:=1 to N do begin       E[i,j]:= (i / j) / N + 0.4;
    E_[i]:= (i / j) / N + 0.4;   end;
```

```
 for i:=1 to N do begin
   F1[i]:=F[10,i];
   F2[i]:=F[1,i];
 end;
 for i:=1 to N do
   Z[i] := abs(F1[i]-F2[i]/E_[i]);
 for i := 1 to N do
     Y[i]:= O[i]+Z[i]+O[i]*F1[i];
 for i:=1 to N do
   for j:=1 to N do
     AdvStringGrid8.Cells[i,j]:=Floattostr(F[i,j]);
 ReadFile();
  for i := 1 to 7 do
    for j:=1 to 4 do form1.AdvStringGrid4.Cells[j-1,i]:='0';
```

arOp[1,1]:='Изложение в законодательных, нормативных и методических документах вопросов, определяющих перечень сведений использующихся в процессах и программах ИС, которые подлежат защите и порядок определения таких сведений.';

arOp[1,2]:='Описание функций органов, ответственных за определение сведений, подлежащих защите при использовании их в процессах и программах ИС.';

arOp[1,3]:='Описание мер (политики безопасности), обеспечивающих своевременное и качественное определение перечня сведений, подлежащих защите при использовании их в процессах и программах ИС.';

arOp[1,4]:='Описание набора средств для обеспечения оперативности и качества определения информации, подлежащей защите при использовании их в процессах и программах ИС.';

arOp[2,1]:='Изложение в законодательных, нормативных и методических документах вопросов, определяющих порядок выявления потенциальных каналов утечки информации в процессах и программах ИС.';

arOp[2,2]:='Описание функций органов, ответственных за выявление потенциальных каналов утечки информации в процессах и программах ИС.';

arOp[2,3]:='Описание мер (политики безопасности), обеспечивающих своевременное и качественное выявление потенциальных каналов утечки информации в процессах и программах ИС.';

arOp[2,4]:='Описание набора средств для обеспечения оперативности и качества выявление потенциальных каналов утечки информации в процессах и программах ИС.';

arOp[3,1]:='Изложение в законодательных, нормативных и методических документах вопросов, определяющих проведение оценки уязвимости и рисков для информации в процессах и программах ИС.';

arOp[3,2]:='Описание функций органов, ответственных за проведение оценки уязвимости и рисков для информации в процессах и программах ИС.';

arOp[3,3]:='Описание мер (политики безопасности), обеспечивающих своевременное и качественное проведение оценки уязвимости и рисков для информации в процессах и программах ИС.';

arOp[3,4]:='Описание набора средств для обеспечения оперативности и качества проведение оценки уязвимости и рисков для информации в процессах и программах ИС.';

arOp[4,1]:='Изложение в законодательных, нормативных и методических документах вопросов определения требований к СЗИ в процессах и программах ИС.';

arOp[4,2]:='Описание функций органов, ответственных за определение требований к СЗИ в процессах и программах ИС.';

arOp[4,3]:='Описание мер (политики безопасности), обеспечивающих своевременное и качественное определение требований к СЗИ в процессах и программах ИС.';

```
 arOp[4,4]:='Описание набора средств  для обеспечения оперативности и качества определения требований к СЗИ в процессах и программах ИС.';
 arOp[5,1]:='Изложение в законодательных, нормативных и методических документах вопросов, определяющих осуществление выбора средств защиты для информации в процессах и программах ИС.';
 arOp[5,2]:='Описание функций органов, ответственных за осуществление выбора средств защиты для информации в процессах и программах ИС.';
 arOp[5,3]:='Описание мер (политики безопасности), обеспечивающих своевременное и качественное осуществление выбора средств защиты для информации в процессах и программах ИС.';
 arOp[5,4]:='Описание набора средств  для обеспечения оперативности и качества осуществление выбора средств защиты для информации в процессах и программах ИС.';
 arOp[6,1]:='Изложение в законодательных, нормативных и методических документах вопросов, определяющих порядок внедрения и использование выбранных мер и средств защиты для информации в процессах и программах ИС.';
 arOp[6,2]:='Описание функций органов, ответственных за внедрение и использование выбранных мер и средств защиты для информации в процессах и программах ИС.';
 arOp[6,3]:='Описание мер (политики безопасности), обеспечивающих своевременное и качественное внедрение и использование выбранных способов и средств защиты для информации в процессах и программах ИС.';
 arOp[6,4]:='Описание набора средств  для обеспечения оперативности и качества внедрения и использования выбранных мер и средств защиты для информации в процессах и программах ИС.';
 arOp[7,1]:='Изложение в законодательных, нормативных и методических документах вопросов, определяющих порядок контроля целостности и управления защитой для информации в процессах и программах ИС.';
 arOp[7,2]:='Описание функций органов, ответственных за осуществление контроля целостности и управление защитой для информации в процессах и программах ИС.';
 arOp[7,3]:='Описание мер (политики безопасности), обеспечивающих своевременный и качественный контроль целостности и управление защитой для информации в процессах и программах ИС.';
 arOp[7,4]:='Описание набора средств  для обеспечения оперативности и качества контроля целостности и управления защитой для информации в процессах и программах ИС.';
end;
procedure TForm1.AdvStringGrid7KeyPress(Sender: TObject; var Key: Char);
begin
  if not (key in ['0'..'9','.']) then
  key := #0;
end;
procedure TForm1.AdvStringGrid8KeyPress(Sender: TObject; var Key: Char);
begin
  if not (key in ['0'..'9','.']) then
  key := #0;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
  ScanLocalNW;
end;
procedure TForm1.TreeView1Click(Sender: TObject);
var   p1,a,b,c,d,p2:integer;
  st:string;
```

```
begin
  if Assigned(TreeView1.Selected) then begin
    StatusBar1.Panels[0].Text := '   ' + TreeView1.Selected.Text;
    st:=TreeView1.Selected.Text;
    p1:=Pos(':',st);
    if p1<>0 then begin
      IPst:=Copy(st,p1+1,length(st)-p1);
      p2:=Pos('.',st);
      a:=StrToInt(Copy(st, p1+1, p2-p1-1));
      st[p2]:=',';
      p1:=p2;
      p2:=Pos('.',st);
      b:=StrToInt(Copy(st, p1+1, p2-p1-1));
      st[p2]:=',';
      p1:=p2;
      p2:=Pos('.',st);
      c:=StrToInt(Copy(st, p1+1, p2-p1-1));
      st[p2]:=',';
      p1:=p2;
      p2:=length(st)-p1;
      d:=StrToInt(Copy(st, p1+1, p2));
      IP := MAKEIPADDRESS(a,b,c,d);
      InfComp;
    end;
  end
  else StatusBar1.Panels[0].Text := STR_FIELD;
end;
procedure TForm1.TreeView1DblClick(Sender: TObject);
var
  Str: String;
begin
  if Assigned(TreeView1.Selected) then
  begin
    Str := TreeView1.Selected.Text;
    if Copy(Str, 1, 2) <> '\\' then Exit;
    if Pos(' IP:', Str) <> 0 then
      ShellExecute(Handle, 'explore', PChar(Copy(Str, 1, Pos(' IP:', Str))), nil, nil, SW_SHOW)
    Else       ShellExecute(Handle, 'explore', PChar(Str), nil, nil, SW_SHOW);
  end; end;

procedure TForm1.btnStartClick(Sender: TObject);
var
  I, AFrom, ATo: Integer;
  Prefix: String;
  function ValidRange: Boolean;
  var     F, T: TInAddr;
  begin
    F.S_addr := inet_addr(PChar(IPFrom.Text));
    T.S_addr := inet_addr(PChar(IPTo.Text));
    Result := (F.S_un_b.s_b1 = T.S_un_b.s_b1) and
              (F.S_un_b.s_b2 = T.S_un_b.s_b2) and
              (F.S_un_b.s_b3 = T.S_un_b.s_b3);
```

```
    if Result then
    begin
      AFrom := Integer(F.S_un_b.s_b4);
      ATo := Integer(T.S_un_b.s_b4);
      Prefix := IntToStr(Integer(F.S_un_b.s_b1)) + '.' +
              IntToStr(Integer(F.S_un_b.s_b2)) + '.' +
              IntToStr(Integer(F.S_un_b.s_b3)) + '.';
      ProgressBar.Max := ATo - AFrom;
      ProgressBar.Position := 0;
    end
    else
      MessageDlg(RES_ERR_RANGE, mtError, [mbOK], 0);
  end;
begin
  CompFound := 0;
  ThreadCount := 0;
  tvResult.Items.Clear;
  if ValidRange then
  begin
    btnStart.Enabled := False;
    for I := AFrom to ATo do
      with TScanThread.Create(False) do
      begin
        IP := inet_addr(PChar(Prefix + IntToStr(I)));
        FreeOnTerminate := True;
        Resume;
      end;   end; end;

constructor TIPEdit.Create(AOwner: TWinControl; Rect: TRect);
begin
  InitCommonControl(ICC_INTERNET_CLASSES);
  FHandle:= CreateWindow(WC_IPADDRESS, nil, WS_CHILD or WS_VISIBLE,
    Rect.Left, Rect.Top, Rect.Right, Rect.Bottom, AOwner.Handle, 0, hInstance, nil);
  FFont := CreateFont(-11, 0, 0, 0, 400, 0, 0, 0, DEFAULT_CHARSET,
    OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY,
    DEFAULT_PITCH or FF_DONTCARE, 'MS Sans Serif');
  SendMessage(FHandle, WM_SETFONT, FFont, 0);
  Text := '0.0.0.0';
end;

destructor TIPEdit.Destroy;
begin   DeleteObject(FFont);
  inherited;
end;
function TIPEdit.GetText: String;
begin
  SendMessage(FHandle, IPM_GETADDRESS, 0, Longint(PDWORD(@FIP)));
  Result := IntToStr(FIRST_IPADDRESS(FIP))+
      '.' + IntToStr(SECOND_IPADDRESS(FIP)) +
      '.' + IntToStr(THIRD_IPADDRESS(FIP)) +
      '.' + IntToStr(FOURTH_IPADDRESS(FIP));
end;
```

```
procedure TIPEdit.SetText(const Value: String);
  function MakeIPAddressEx(b1, b2, b3, b4: Char):LPARAM;
  begin
    Result := MAKEIPADDRESS(DWORD(b1), DWORD(b2), DWORD(b3), DWORD(b4));
  end;
var
  Tmp: TInAddr;
begin
  Tmp.S_addr := inet_addr(PChar(Value));
  if Tmp.S_addr = INADDR_NONE then Exit;
  with Tmp.S_un_b do
    FIP := MakeIPAddressEx(s_b1, s_b2, s_b3, s_b4);
  SendMessage(FHandle, IPM_SETADDRESS, 0, FIP);
end;
{TScanThread }
procedure TScanThread.DecCount;
begin
  Form1.ThreadCount := Form1.ThreadCount - 1;
end;
procedure TScanThread.Execute;
begin
  inherited;
  Synchronize(IncCount);
  Scan;
  Synchronize(DecCount);
end;
function TScanThread.GetCompName(const Addr: Integer): String;
var
  WSA: TWSAData;
  Host: PHostEnt;
  Err: Integer;
begin
  Result := RES_UNKNOWN;
  Err := WSAStartup(WSA_TYPE, WSA);
  if Err <> 0 then  // Лучше пользоваться такой конструкцией,
  begin             // чтобы в случае ошибки можно было увидеть ее код.
    //ShowMessage(SysErrorMessage(GetLastError));
    Exit;
  end;
  try
    if Addr = INADDR_NONE then Exit;
    Host := gethostbyaddr(@Addr, SizeOf(Addr), PF_INET);
    if Assigned(Host) then  // Обязательная проверка, в противном случае, при
      Result := Host.h_name // отсутствии компьютера с заданым IP, получим AV
    else
  finally
    WSACleanup;
  end; end;
procedure TScanThread.IncCount;
begin
  Form1.ThreadCount := Form1.ThreadCount + 1;
end;
```

```
procedure TScanThread.Scan;
type
  TShareInfo1Array = array of TShareInfo1;
var
  entriesread, totalentries: DWORD;
  Info: Pointer;
  I: Integer;
  CompName: PWideChar;
begin
  CompName := StringToOleStr(GetCompName(FIP));
  if CompName = RES_UNKNOWN then Exit;
  FRes := TStringList.Create;
  try
    Fres.Add(CompName);
    if NetShareEnum(CompName, 1, Info, DWORD(-1), @entriesread,
      @totalentries, nil) = 0 then
    try
     if entriesread > 0 then
     begin
       for I := 0 to entriesread - 1 do
         FRes.Add(TShareInfo1Array(@(Info^))[I].shi1_netname);
       Synchronize(UpdateTree);
     end;
    finally
     NetApiBufferFree(Info);
    end;
  finally
    FRes.Free;
  end; end;

procedure TScanThread.UpdateTree;
var
  I: Integer;
  Root: TTreeNode;
begin
  Form1.tvResult.Items.BeginUpdate;
  try
    Root := Form1.tvResult.Items.Add(nil, FRes.Strings[0]);
    for I := 1 to FRes.Count - 1 do
      Form1.tvResult.Items.AddChild(Root, FRes.Strings[I]);
      Form1.CompFound := Form1.CompFound + 1;
  finally
    Form1.tvResult.Items.EndUpdate;
  end; end;
procedure InfComp;
var
  TmpCompName, TmpProvider, TmpGroup, TmpUser, TmpServer: String;
  Time: Cardinal;
  IPStr: String;
begin
 with Form1 do begin
    Time := GetTickCount;  // Засечем время...
```

```
   // Преобразуем эту абракадабру в нормальный "Dotted IP"
   IPStr := IntToStr(FIRST_IPADDRESS(IP));
   IPStr := IPStr + '.' + IntToStr(SECOND_IPADDRESS(IP));
   IPStr := IPStr + '.' + IntToStr(THIRD_IPADDRESS(IP));
   IPStr := IPStr + '.' + IntToStr(FOURTH_IPADDRESS(IP));
   with memInfo, memInfo.Lines do begin                    // Вывод информации
    Clear;                                  // Очищаем экран
    Refresh;                                 // Обновляем...
                                          // (при вызове первой функции может не обновиться)
    Add(RES_IP + IPStr);                          // Выводим IP адрес
    TmpCompName := GetNameFromIP(IPStr);
    if TmpCompName = RES_UNKNOWN then Exit;
    Add(RES_CMP + TmpCompName);                    // Выводим имя компьютера
    TmpUser := GetUsers(IPStr);
    Add(RES_USR + TmpUser);                       // Выводим имя пользователя
    TmpProvider := GetProvider(TmpCompName);
    Add(RES_PROV + TmpProvider);                    // Выводим провайдера
    Add(RES_COM + GetComment(TmpCompName,
     TmpProvider));                               // Выводим комментарий к ресурсу
    TmpGroup := GetDomain(TmpCompName, TmpProvider);
    Add(RES_DOM + TmpGroup);                       // Выводим группу
    TmpServer := GetDomainServer(TmpGroup);
    if TmpServer <> '' then begin
     Add(RES_SER + TmpServer);                     // Выводим имя сервера
     Add(RES_GRP + GetGroups(TmpServer, TmpUser));      // Выводим группы домена в
которые входит пользователь
    end;
    Add(RES_SHARES + GetShares(TmpCompName));          // Выводим список доступных
ресурсов
    Add(RES_MAC + GetMacFromIP(IPStr));               // Выводим MAC адрес
    Add(RES_TIME + IntToStr(GetTickCount - Time));     // Сколько времени затрачено
  end;   end; end;

function TForm1.GetNameFromIP(const IP: String): String;
var
 WSA: TWSAData;
 Host: PHostEnt;
 Addr: Integer;
 Err: Integer;
begin
 Result := RES_UNKNOWN;
 Err := WSAStartup(WSA_TYPE, WSA);
 if Err <> 0 then  // Лучше пользоваться такой конструкцией,
 begin           // чтобы в случае ошибки можно было увидеть ее код.
  //ShowMessage(SysErrorMessage(GetLastError));
  Exit;
 end;
 try
  Addr := inet_addr(PChar(IP));
  if Addr = INADDR_NONE then
  begin
   //ShowMessage(SysErrorMessage(GetLastError));
```

```
      WSACleanup;
      Exit;
    end;
    Host := gethostbyaddr(@Addr, SizeOf(Addr), PF_INET);
    if Assigned(Host) then  // Обязательная проверка, в противном случае, при
      Result := Host.h_name // отсутствии компьютера с заданым IP, получим AV
    else
      //ShowMessage(SysErrorMessage(GetLastError));
  finally
    WSACleanup;
  end; end;
// Перечисляем всех залогиненных на машине пользователей
// начинаем перечисления со второго пользователя, потому,  что
// первым будет "имя компьютера"$
function TForm1.GetUsers(const CompName: String): String;
var
  Buffer, tmpBuffer: Pointer;
  PrefMaxLen       : DWORD;
  Resume_Handle    : DWORD;
  EntriesRead      : DWORD;
  TotalEntries     : DWORD;
  I, Size          : Integer;
  PSrvr            : PWideChar;
begin
  PSrvr := nil;
  try
    // Переводим имя компьютера типа PWideChar
    Size := Length(CompName);
    GetMem(PSrvr, Size * SizeOf(WideChar) + 1);
    StringToWideChar(CompName, PSrvr, Size + 1);
    PrefMaxLen := DWORD(-1);
    EntriesRead := 0;
    TotalEntries := 0;
    Resume_Handle := 0;
    Buffer := nil;
    // Получаем список пользователей на компьютере из PSrvr
    if NetWkstaUserEnum( PSrvr, 1, @Buffer, PrefMaxLen, @EntriesRead,
      @TotalEntries, @Resume_Handle) = S_OK then
    begin
      tmpBuffer := Pointer(DWORD(Buffer) + SizeOf(WKSTA_USER_INFO_1));
      for I := 1 to TotalEntries - 1 do
      begin
        Result := Result + WKSTA_USER_INFO_1(tmpBuffer^).wkui1_username + ', ';
        tmpBuffer := Pointer(DWORD(tmpBuffer) + SizeOf(WKSTA_USER_INFO_1));
      end;
      Result := Copy(Result, 1, Length(Result) - 2);
  end
  else     //ShowMessage(SysErrorMessage(GetLastError));
  finally
    NetApiBufferFree(Buffer);
    FreeMem(PSrvr);
  end; end;
```

```
function TForm1.GetDomain(const CompName, Provider: String): String;
var
  CurrRes: TNetResource;
  ParentName: array [0..1] of TNetResource;
  Enum: DWORD;
  Err: Integer;
begin
  with CurrRes do
  begin
    dwScope := RESOURCE_GLOBALNET;
    dwType := RESOURCETYPE_DISK;
    dwDisplayType := RESOURCEDISPLAYTYPE_SERVER;
    dwUsage := RESOURCEUSAGE_CONTAINER;
    lpLocalName := '';
    lpRemoteName := PChar('\\' + CompName);
    lpComment := '';
    lpProvider := PChar(Provider);
  end;
  Enum := SizeOf(ParentName);
  Err := WNetGetResourceParent(@CurrRes, @ParentName, Enum);
  if Err = NO_ERROR then
  begin
    Result := ParentName[0].lpRemoteName;
    if Result = '' then Result := RES_COM_NO;
  end
  else
    //ShowMessage(SysErrorMessage(GetLastError));
end;
function TForm1.GetComment(CompName, Provider: String): String;
var
  StopScan: Boolean;
  TmpRes: TNetResource;
  // Сканирование
  procedure Scan(Res: TNetResource; Root: boolean);
  var
    Enum, I: Cardinal;
    ScanRes: array [0..512] of TNetResource; // Можно сделать и больший размер массива
    Size, Entries, Err: DWORD;               // но, как показывает практика, такого достаточно
  begin
      if StopScan then Exit; // Используем флаг для выхода из рекурсии
    if Root = True then
      Err := WNetOpenEnum(RESOURCE_GLOBALNET, RESOURCETYPE_DISK,
        0, nil, Enum) // корневой...
    else
      Err := WNetOpenEnum(RESOURCE_GLOBALNET, RESOURCETYPE_DISK,
        0, @Res, Enum); // и рекурсионный для поиска вложений...
    if Err = NO_ERROR then
    begin
      Size := SizeOf(ScanRes);
      Entries := DWORD(-1);
      Err := WNetEnumResource(Enum, Entries, @ScanRes, Size);
      if Err = NO_ERROR then
```

```
      try
       for I := 0 to Entries - 1 do
       begin
        if StopScan then Exit; // Еще один флаг, так как выход на верхний вызов
        with ScanRes[i] do     // может осуществиться из цикла
        begin
         if dwDisplayType = RESOURCEDISPLAYTYPE_SERVER then
          if lpRemoteName = CompName then // если нашли наш компьютер...
          begin
           Result :=  lpComment;
           StopScan := True;        // Выставляем флаг для выхода из рекуссии
           Exit;
          end;
         if dwDisplayType <> RESOURCEDISPLAYTYPE_SERVER then    // не будем
сканировать шары у компов...
           Scan(ScanRes[i], False);
        end;
       end;
      finally
       WNetCloseEnum(Enum);
      end
      else
       if Err <> ERROR_NO_MORE_ITEMS then  // Нет элементов для отображения...
        MessageDlg(SysErrorMessage(GetLastError), mtError, [mbOK], 0);
    end
    else   end;
// Основная процедура
begin
  // Подготовительные действия...
  Result := RES_UNKNOWN;
  if CompName = RES_UNKNOWN then Exit;    // Если имя компа не найдено,
                                  // незачем и продолжать.
  CompName := '\\' + CompName;            // Подправим имя,
                                  // чтоб не делать это далее в цикле...

  StopScan := False;    // Снимем флаг выхода из рекурсии.
                  // Запускаем сканирование...
  Scan(TmpRes, True);
  // Результаты...
  if Result = '' then Result := RES_COM_NO;
end;
function TForm1.GetProvider(const CompName: String): String;
var
  Buffer: array [0..255] of Char;
  Size: DWORD;
begin
  Size := SizeOf(Buffer);
  if WNetGetProviderName(WNNC_NET_LANMAN, @Buffer, Size) <> NO_ERROR then
    Result := RES_COM_NO
  else
    Result := String(Buffer);
end;
```

```
function TForm1.GetMacFromIP(const IP: String): String;
  function GetMAC(Value: TMacAddress; Length: DWORD): String;
  var
    I: Integer;
  begin
    if Length = 0 then Result := '00-00-00-00-00-00' else
    begin
      Result := '';
      for i:= 0 to Length -2 do
        Result := Result + IntToHex(Value[i], 2) + '-';
      Result := Result + IntToHex(Value[Length-1], 2);
    end;
  end;
  // Получаем IP адрес, заметь в отличии от работы с классом WC_IPADDRESS
  // здесь преобразование идет в обратном порядке!
  function GetDottedIPFromInAddr(const InAddr: Integer): String;
  begin     Result := '';
    Result := IntToStr(FOURTH_IPADDRESS(InAddr));
    Result := Result + '.' + IntToStr(THIRD_IPADDRESS(InAddr));
    Result := Result + '.' + IntToStr(SECOND_IPADDRESS(InAddr));
    Result := Result + '.' + IntToStr(FIRST_IPADDRESS(InAddr));
  end;

  // Основная функция
var
  Table: TMibIPNetTable;
  Size: Integer;
  CatchIP: String;
  Err, I: Integer;
Begin   Result := RES_UNKNOWN;
  Size := SizeOf(Table);
  Err := GetIpNetTable(@Table, @Size, False);
  if Err <> NO_ERROR then
  begin
    Exit;
  end;
  // Теперь мы имеем таблицу из IP адресов и соответсвующих им MAC адресов
  for I := 0 to Table.dwNumEntries - 1 do     // Ищем нужный IP ...
  begin
    CatchIP := GetDottedIPFromInAddr(Table.Table[I].dwAddr);
    if CatchIP = IP then                      // И выводим его MAC ...
    begin
      Result := GetMAC(Table.Table[I].bPhysAddr, Table.Table[I].dwPhysAddrLen);
      Break;     end;   end; end;
// Получение доступных сетевых ресурсов на удаленном компьютере
function TForm1.GetShares(const CompName: String): String;
type TShareInfo1Array = array of TShareInfo1;
var
  entriesread, totalentries: DWORD;
  Info: Pointer;
  I: Integer;
  CN: PWideChar;
```

```
begin
  CN := StringToOleStr(CompName);
  // так как нам нужны только имена ресурсов, воспользуемся струтурой TShareInfo1
  // тогда, не нужно будет получать привилегии администратора на удаленной машине :)
  if NetShareEnum(CN, 1, Info, DWORD(-1), @entriesread,
    @totalentries, nil) = 0 then
    try // список ресурсов смотрим здесь
      if entriesread > 0 then
        for I := 0 to entriesread - 1 do
          Result := Result + TShareInfo1Array(@(Info^))[I].shi1_netname + ' ';
    finally
      NetApiBufferFree(Info);
    end; end;

// Имя сервера домена
function TForm1.GetDomainServer(const DomainName: String): String;
var
  Domain: PWideChar;
  Server: PWideChar;
begin
  GetMem(Domain, MAX_PATH);
  try
    StringToWideChar(DomainName, Domain, MAX_PATH);
    if NetGetAnyDCName(nil, Domain, @Server)= NO_ERROR then
    try       Result := WideCharToString(Server);
              finally
      NetApiBufferFree(Server);
            end;
  finally
    FreeMem(Domain, MAX_PATH);
  end; end;
// перечисление доменных групп в которые входит пользователь
function TForm1.GetGroups(DomainServer: String; UserName: String): String;
type
  TGroupUsersInfoArray = array of TGroupUsersInfo0;
var
  Info: PGroupUsersInfo0;
  Sn, Un: PWideChar;
  entriesread, totalentries: DWORD;
  I, A, B, Size: Integer;
  P: Pointer;
begin
  // Имя сервера домена
  Sn := StringToOLEStr(DomainServer);
  // Имя пользователя
  Un := StringToOleStr(UserName);
  // делаем запрос
  if NetUserGetGroups(Sn, Un, 0, @Info, DWORD(-1), entriesread, totalentries) = NO_ERROR
then
  try
    if entriesread > 0 then
      for I := 0 to entriesread - 1 do
```

```
      Result := Result + TGroupUsersInfoArray(@(Info^))[I].grui0_name + ' ';
  finally
    NetApiBufferFree(Info);
  end;
end;
// Функция преобразует состояние порта в строковый эквивалент
function TForm1.PortStateToStr(const State: DWORD): String;
begin
  case State of
    MIB_TCP_STATE_CLOSED: Result := 'CLOSED';
    MIB_TCP_STATE_LISTEN: Result := 'LISTEN';
    MIB_TCP_STATE_SYN_SENT: Result := 'SYN SENT';
    MIB_TCP_STATE_SYN_RCVD: Result := 'SYN RECEIVED';
    MIB_TCP_STATE_ESTAB: Result := 'ESTABLISHED';
    MIB_TCP_STATE_FIN_WAIT1: Result := 'FIN WAIT 1';
    MIB_TCP_STATE_FIN_WAIT2: Result := 'FIN WAIT 2';
    MIB_TCP_STATE_CLOSE_WAIT: Result := 'CLOSE WAIT';
    MIB_TCP_STATE_CLOSING: Result := 'CLOSING';
    MIB_TCP_STATE_LAST_ACK: Result := 'LAST ACK';
    MIB_TCP_STATE_TIME_WAIT: Result := 'TIME WAIT';
    MIB_TCP_STATE_DELETE_TCB: Result := 'DELETE TCB';
  else
    Result := 'UNKNOWN';
  end; end;
procedure TForm1.GetMemoryInfo;
var
 MemInfo : TMemoryStatus;
begin
 MemInfo.dwLength := Sizeof (MemInfo);
 GlobalMemoryStatus (MemInfo);
 TotalPhys.caption:=inttostr(MemInfo.dwTotalPhys div 1048576) + ' Mb';
 AvailPhys.caption:=inttostr(MemInfo.dwAvailPhys div 1048576) + ' Mb';
 TotalPage.caption:=inttostr(MemInfo.dwTotalPageFile div 1048576) + ' Mb';
 AvailPage.caption:=inttostr(MemInfo.dwAvailPageFile div 1048576) + ' Mb';
 AdvProgressBar1.Position := MemInfo.dwAvailPhys div (MemInfo.dwTotalPhys div 100);
 AdvProgressBar2.Position := MemInfo.dwAvailPageFile div (MemInfo.dwTotalPageFile div
100);
end;
procedure TForm1.GetCompInfo;
var
 SystemIniFile:TIniFile;
 RegFile:TRegIniFile;
 PathArray : array [0..255] of char;
 OSVersion: TOSVersionInfo;
begin
 //Computer
 SystemIniFile:=TIniFile.Create('\\'+IPst+'\System.ini');
 ComputerLabel.Caption:=SystemIniFile.ReadString('boot.description', 'system.drv', 'Unknown');
 SystemIniFile.Free;
 RegFile:=TRegIniFile.Create('Software');
 RegFile.RootKey:=HKEY_LOCAL_MACHINE;
 RegFile.OpenKey('hardware',false);
```

```
 RegFile.OpenKey('DESCRIPTION',false);
 RegFile.OpenKey('System',false);
 RegFile.OpenKey('CentralProcessor',false);
 ProcessorLabel.Caption:=RegFile.ReadString('0','Identifier','Unknown');
 MMXIdentifierLabel.Caption:=RegFile.ReadString('0','MMXIdentifier','Unknown');
 VendorIdentifierLabel.Caption:=RegFile.ReadString('0','VendorIdentifier','Unknown');
//OS
 OSVersion.dwOSVersionInfoSize := SizeOf(OSVersion);
 if GetVersionEx(OSVersion) then
  begin
   VersionLabel.Caption:= Format('%d.%d (%d.%s)',[OSVersion.dwMajorVersion,
OSVersion.dwMinorVersion,(OSVersion.dwBuildNumber and $FFFF),
OSVersion.szCSDVersion]);
   case OSVersion.dwPlatformID of
    VER_PLATFORM_WIN32s:       VersionNumberLabel.Caption := 'Windows 3.1';
    VER_PLATFORM_WIN32_WINDOWS: VersionNumberLabel.Caption := 'Windows 95';
    VER_PLATFORM_WIN32_NT:     VersionNumberLabel.Caption := 'Windows NT';
   else                    VersionNumberLabel.Caption := '';
   end;   end;
 RegFile.CloseKey;
 RegFile.OpenKey('SOFTWARE',false);
 RegFile.OpenKey('Microsoft',false);
 RegFile.OpenKey('Windows',false);
 OSNameLabel.Caption:=RegFile.ReadString('CurrentVersion','ProductName','Unknown');
RegisteredOrganizationLabel.Caption:=RegFile.ReadString('CurrentVersion','RegisteredOrganiz
ation','Unknown');
RegisteredOwnerLabel.Caption:=RegFile.ReadString('CurrentVersion','RegisteredOwner','Unkn
own');
 SerNumberEdit.Caption:=RegFile.ReadString('CurrentVersion','ProductId','Unknown');
 RegFile.Free;
 FillChar(PathArray, SizeOf(PathArray), #0);
 GetWindowsDirectory(PathArray,255);
 WindowsDirLabel.Caption:= Format('%s',[PathArray]);
 FillChar(PathArray, SizeOf(PathArray), #0);
 ExpandEnvironmentStrings('%TEMP%', PathArray, 255);
 TempDir.Caption:=Format('%s',[PathArray]);
end;
procedure TForm1.GetIPInfo;
var
 FixedInfoSize, Err, AdapterInfoSize:DWORD;
 pFixedInfo:PFIXED_INFO;
 pAdapterInfo, pAdapt:PIP_ADAPTER_INFO;
 pAddrStr:PIP_ADDR_STRING;
begin
 FixedInfoSize:=0;
 Err:=GetNetworkParams(nil, FixedInfoSize);
 if (Err<>0) and (Err<>ERROR_BUFFER_OVERFLOW) then
  begin
   HostNameLabel.Caption:='Error';
   exit;   end;
 pFixedInfo:=PFIXED_INFO(GlobalAlloc(GPTR, FixedInfoSize));
 GetNetworkParams(pFixedInfo, FixedInfoSize);
```

```
HostNameLabel.Caption:=StrPas(pFixedInfo.HostName);
DNSListBox.Items.Clear;
DNSListBox.Items.Add(StrPas(pFixedInfo.DnsServerList.IpAddress.S));
pAddrStr:=pFixedInfo.DnsServerList.Next;
while (pAddrStr<>nil) do
 begin
   DNSListBox.Items.Add(StrPas(pAddrStr.IpAddress.S));
  pAddrStr:=pAddrStr.Next;
 end;
case pFixedInfo.NodeType of
 1: NodeTypeLabel.Caption:='Broadcast';
 2: NodeTypeLabel.Caption:='Peer to peer';
 4: NodeTypeLabel.Caption:='Mixed';
 8: NodeTypeLabel.Caption:='Hybrid';
end;
NetBIOSScopeLabel.Caption:=pFixedInfo.ScopeId;
if pFixedInfo.EnableRouting>0 then
 IPRoutingLabel.Caption:='Yes'
else
 IPRoutingLabel.Caption:='No';
if pFixedInfo.EnableProxy>0 then
 WINSProxyLabel.Caption:='Yes'     else
 WINSProxyLabel.Caption:='No';
if pFixedInfo.EnableDns>0 then
 NetBIOSResolutionLabel.Caption:='Yes'
else
 NetBIOSResolutionLabel.Caption:='No';
//Get Adapter Info
AdapterCB.Items.Clear;
AdapterInfoSize:=0;
Err:=GetAdaptersInfo(nil, AdapterInfoSize);
if (Err<>0) and (Err<>ERROR_BUFFER_OVERFLOW) then
 begin
  AdapterCB.Items.Add('Error');
  exit;
 end;
pAdapterInfo := PIP_ADAPTER_INFO(GlobalAlloc(GPTR, AdapterInfoSize));
GetAdaptersInfo(pAdapterInfo, AdapterInfoSize);
pAdapt := pAdapterInfo;
while pAdapt<>nil do
 begin
  case pAdapt.Type_ of
   MIB_IF_TYPE_ETHERNET:
    AdapterCB.Items.Add('Ethernet adapter '+pAdapt.AdapterName);
   MIB_IF_TYPE_TOKENRING:
    AdapterCB.Items.Add('Token Ring adapter '+pAdapt.AdapterName);
   MIB_IF_TYPE_FDDI:
    AdapterCB.Items.Add('FDDI adapter '+pAdapt.AdapterName);
   MIB_IF_TYPE_PPP:
    AdapterCB.Items.Add('PPP adapter '+pAdapt.AdapterName);
   MIB_IF_TYPE_LOOPBACK:
    AdapterCB.Items.Add('Loopback adapter '+pAdapt.AdapterName);
```

```
    MIB_IF_TYPE_SLIP:
     AdapterCB.Items.Add('Slip adapter '+pAdapt.AdapterName);
    MIB_IF_TYPE_OTHER:
     AdapterCB.Items.Add('Other adapter '+pAdapt.AdapterName);
   end;
   pAdapt := pAdapt.Next;
  end;
 GlobalFree(Cardinal(pFixedInfo));
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
 PageControl3.ActivePageIndex:=0;
 GetMemoryInfo;
 GetCompInfo;
 GetIPInfo;
 GetAdapterInfo;
 UpdateDisk;
end;
procedure TForm1.GetAdapterInfo;
var
 Err, AdapterInfoSize:DWORD;
 pAdapterInfo, pAdapt:PIP_ADAPTER_INFO;
 Str:String;
 i:Integer;
 pAddrStr:PIP_ADDR_STRING;
begin
 AdapterTypeLabel.Caption:='';
 AdapterNameLabel.Caption:='';
 DescriptionLabel.Caption:='';
 PhysicaladdressLabel.Caption:='';
 IPListView.Clear;
 GatewayLabel.Caption:='';
 DHCPLabel.Caption:='';
 DHCPServerLabel.Caption:='';
 SecondaryWINSLabel.Caption:='';
 PrimaryWINSLabel.Caption:='';
 AdapterInfoSize:=0;
 Err:=GetAdaptersInfo(nil, AdapterInfoSize);
 if (Err<>0) and (Err<>ERROR_BUFFER_OVERFLOW) then
  begin
   AdapterCB.Items.Add('Error');
   exit;   end;
 pAdapterInfo := PIP_ADAPTER_INFO(GlobalAlloc(GPTR, AdapterInfoSize));
 GetAdaptersInfo(pAdapterInfo, AdapterInfoSize);
 pAdapt := pAdapterInfo;
 while pAdapt<>nil do
  begin
   case pAdapt.Type_ of
    MIB_IF_TYPE_ETHERNET:
     Str:='Ethernet adapter ';
    MIB_IF_TYPE_TOKENRING:
     Str:='Token Ring adapter ';
```

```
    MIB_IF_TYPE_FDDI:
     Str:='FDDI adapter ';
    MIB_IF_TYPE_PPP:
     Str:='PPP adapter ';
    MIB_IF_TYPE_LOOPBACK:
     Str:='Loopback adapter ';
    MIB_IF_TYPE_SLIP:
     Str:='Slip adapter ';
    MIB_IF_TYPE_OTHER:
     Str:='Other adapter ';
   end;
   if Str+pAdapt.AdapterName<>AdapterCB.Text then
    begin
     pAdapt := pAdapt.Next;
     Continue;
    end;
    AdapterTypeLabel.Caption:=Str;
    AdapterNameLabel.Caption:=AdapterCB.Text;
    DescriptionLabel.Caption:=pAdapt.Description;
    Str:='';
    for i:=0 to pAdapt.AddressLength-1 do
     begin
      Str:=Str+IntToHex(pAdapt.Address[i],2);
      if i<>Integer(pAdapt.AddressLength-1) then
       Str:=Str+'-';
     end;
    PhysicaladdressLabel.Caption:=Str;
    pAddrStr:=@pAdapt.IpAddressList;
    while pAddrStr<>nil do
     begin
      with IPListView.Items.Add do
       begin
        Caption:=pAddrStr.IpAddress.S;
        SubItems.Add(pAddrStr.IpMask.S);
       end;
      pAddrStr := pAddrStr.Next;
     end;
    if pAdapt.DhcpEnabled=0 then
     DHCPLabel.Caption:='no'
    else
     DHCPLabel.Caption:='yes';
    DHCPServerLabel.Caption:=pAdapt.DhcpServer.IpAddress.S;
    PrimaryWINSLabel.Caption:=pAdapt.PrimaryWinsServer.IpAddress.S;
    SecondaryWINSLabel.Caption:=pAdapt.SecondaryWinsServer.IpAddress.S;
    GatewayLabel.Caption:=pAdapt.GatewayList.IpAddress.S;
    break;
  end;
 GlobalFree(Cardinal(pAdapterInfo));
end;
procedure TForm1.UpdateDisk;
var
 lpRootPathName          : PChar;
```

```
 lpVolumeNameBuffer       : PChar;
 nVolumeNameSize          : DWORD;
 lpVolumeSerialNumber     : DWORD;
 lpMaximumComponentLength : DWORD;
 lpFileSystemFlags        : DWORD;
 lpFileSystemNameBuffer   : PChar;
 nFileSystemNameSize      : DWORD;
 FSectorsPerCluster: DWORD;
 FBytesPerSector   : DWORD;
 FFreeClusters     : DWORD;
 FTotalClusters    : DWORD;
begin
 lpVolumeNameBuffer      := '';
 lpVolumeSerialNumber    := 0;
 lpMaximumComponentLength:= 0;
 lpFileSystemFlags       := 0;
 lpFileSystemNameBuffer  := '';
 try
  GetMem(lpVolumeNameBuffer, MAX_PATH + 1);
  GetMem(lpFileSystemNameBuffer, MAX_PATH + 1);
  nVolumeNameSize := MAX_PATH + 1;
  nFileSystemNameSize := MAX_PATH + 1;
  lpRootPathName := PChar(DriveComboBox1.Drive+':\');
  if GetVolumeInformation( lpRootPathName, lpVolumeNameBuffer,
     nVolumeNameSize, @lpVolumeSerialNumber, lpMaximumComponentLength,
     lpFileSystemFlags, lpFileSystemNameBuffer, nFileSystemNameSize )
   then
    begin
     VolumeName.Caption    := lpVolumeNameBuffer;
     VolumeSerial.Caption       :=    IntToHex(HIWord(lpVolumeSerialNumber),    4)    +    '-'    +
IntToHex(LOWord(lpVolumeSerialNumber), 4);
     FileSystemName.Caption:= lpFileSystemNameBuffer;
     GetDiskFreeSpace(            PChar(DriveComboBox1.Drive+':\'),            FSectorsPerCluster,
FBytesPerSector,  FFreeClusters, FTotalClusters);
    end;
 finally
  FreeMem(lpVolumeNameBuffer);
  FreeMem(lpFileSystemNameBuffer);
  end;
 SectorsPerCluster.Caption:=IntToStr(FSectorsPerCluster);
 BytesPerSector.Caption:=IntToStr(FBytesPerSector);
end;
procedure TForm1.DriveComboBox1Change(Sender: TObject);
begin
  UpdateDisk;
end;
procedure TForm1.AdapterCBChange(Sender: TObject);
begin
  GetAdapterInfo;
end;
procedure TForm1.FormShow(Sender: TObject);
begin
```

```
  PageControl3.ActivePageIndex:=0;
  GetMemoryInfo;
  GetCompInfo;
  GetIPInfo;
  GetAdapterInfo;
  UpdateDisk;
end;
// Получение TCP/UDP статистики при помощи стандартных методов
procedure TForm1.Button31Click(Sender: TObject);
var
  Size: DWORD;
  TCPTable: PTMibTCPTable;
  UDPTable: PTMibUdpTable;
  I: DWORD;
begin
  Memo1.Clear;
  // для успешного получения стстаистики первоначально необходимо определиться
  // сколько памяти потребует данная операция
  // Вделяем память под TCP таблицу (под один элемент)
  GetMem(TCPTable, SizeOf(TMibTCPTable));
  try
    // Показываем что памяти у нас не выделено
    Size := 0;
    // Выполняем функцию и после этого переменная Size
    // будет содержать кол-во необходимой памяти
    if GetTcpTable(TCPTable, Size, True) <> ERROR_INSUFFICIENT_BUFFER then Exit;
  finally
    FreeMem(TCPTable);
  end;
  // Теперь выделяем уже требуемое кол-во памяти
  GetMem(TCPTable, Size);
  try
    // Выполняем функцию
    if GetTcpTable(TCPTable, Size, True) = NO_ERROR then
    begin
      Memo1.Lines.Add('');
      Memo1.Lines.Add('            АНАЛИЗ ПО TCP ПРОТОКОЛУ');
      Memo1.Lines.Add('');
      Memo1.Lines.Add(Format('%15s: | %5s %-12s', ['Хост', 'Порт', 'Состояние']));
      Memo1.Lines.Add('=================================================');
    // и насинаем выводить данные по TCP
    for I := 0 to TCPTable^.dwNumEntries - 1 do
      Memo1.Lines.Add(Format('%15s:                    |                %5d                %s',
[inet_ntoa(in_addr(TCPTable^.Table[I].dwLocalAddr)),
       htons(TCPTable^.Table[I].dwLocalPort), PortStateToStr(TCPTable^.Table[I].dwState)]));
    end;
  finally
    FreeMem(TCPTable);
  end;
  // По аналогии поступаем и с UDP статистикой
  GetMem(UDPTable, SizeOf(TMibUDPTable));
  try
```

```
    Size := 0;
    if GetUdpTable(UDPTable, Size, True) <> ERROR_INSUFFICIENT_BUFFER then Exit;
   finally
    FreeMem(UDPTable);
   end;
   GetMem(UDPTable, Size);
   try
    if GetUdpTable(UDPTable, Size, True) = NO_ERROR then
    begin
     Memo1.Lines.Add('');
     Memo1.Lines.Add('            АНАЛИЗ ПО UDP ПРОТОКОЛУ');
     Memo1.Lines.Add('');
     Memo1.Lines.Add(Format('%15s: | %5s ', ['Хост', 'Порт']));
     Memo1.Lines.Add('=======================================');
    for I := 0 to UDPTable^.dwNumEntries - 1 do
     Memo1.Lines.Add(Format('%15s:                                    |                                %5d',
[inet_ntoa(in_addr(UDPTable^.Table[I].dwLocalAddr)),
      htons(UDPTable^.Table[I].dwLocalPort)]));
    end;
   finally
    FreeMem(UDPTable);
    Memo1.Lines.Delete(0);
    Memo1.Lines.Insert(0,'');
   end; end;
// Получение TCP/UDP статистики при помощи недокументрированных методов
procedure TForm1.Button21Click(Sender: TObject);
  // данная функция ищет процесс с th32ProcessID совпадающий с ProcessId
  function ProcessPIDToName(const hProcessSnap: THandle; ProcessId: DWORD): String;
  var
    processEntry: TProcessEntry32;
  begin
    Result := '';
    FillChar(processEntry, SizeOf(TProcessEntry32), #0);
    processEntry.dwSize := SizeOf(TProcessEntry32);
    if not Process32First(hProcessSnap, processEntry) then Exit;
    repeat
      if processEntry.th32ProcessID = ProcessId then
      begin
        // Если нашли нужный процесс - выводим результат и выходим
        Result := String(processEntry.szExeFile);
        Exit;       end;
    // ищем пока не кончатся процессы
    until not Process32Next(hProcessSnap, processEntry);
  end;
var
  TCPExTable: PTMibTCPExTable;
  UDPExTable: PTMibUdpExTable;
  I: DWORD;
  hProcessSnap: THandle;
begin
  Memo1.Clear;
  // для определения каким процессом открыт тот или иной порт
```

```
  // получаем список процессов
  hProcessSnap := CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);
  if (hProcessSnap = INVALID_HANDLE_VALUE) then
  begin
    Memo1.Lines.Add('');
    Memo1.Lines.Add('CreateToolhelp32Snapshot failed');
    Exit;
  end;
  try
    if AllocateAndGetTcpExTableFromStack(@TCPExTable, False, GetProcessHeap, 2, 2) =
NO_ERROR then
    try
      Memo1.Lines.Add('');
      Memo1.Lines.Add('            АНАЛИЗ ПО TCP ПРОТОКОЛУ');
      Memo1.Lines.Add('');
      Memo1.Lines.Add(Format('%15s: | %5s | %-12s | %20s | (%s)', ['Хост', 'Порт', 'Состояние',
'Имя процесса', 'ИД']));
      Memo1.Lines.Add('==================================================');
      // начинаем выводить информацию
      for I := 0 to TCPExTable^.dwNumEntries - 1 do
        Memo1.Lines.Add(Format('%15s: | %5d | %-12s | %20s | (%d)',
          [inet_ntoa(in_addr(TCPExTable^.Table[I].dwLocalAddr)),
          htons(TCPExTable^.Table[I].dwLocalPort),
          PortStateToStr(TCPExTable^.Table[I].dwState),
          // Вот здесь у нас происходит сопоставление процесса открытому порту
          ProcessPIDToName(hProcessSnap, TCPExTable^.Table[I].dwProcessID),
          TCPExTable^.Table[I].dwProcessID]));
    finally
      // Освобождаем память занятую функцией
      GlobalFreePtr(TCPExTable);
    end;
    // По аналогии поступаем и с UDP статистикой
    if AllocateAndGetUdpExTableFromStack(@UDPExTable, False, GetProcessHeap, 2, 2) =
NO_ERROR then
    try
      Memo1.Lines.Add('');
      Memo1.Lines.Add('            АНАЛИЗ ПО UDP ПРОТОКОЛУ');
      Memo1.Lines.Add('');
      Memo1.Lines.Add(Format('%15s: | %5s |  %19s | (%s)', ['Хост', 'Порт',  'Имя процесса',
'ИД']));
      Memo1.Lines.Add('==================================================');
      // начинаем выводить информацию
      for I := 0 to UDPExTable^.dwNumEntries - 1 do
        Memo1.Lines.Add(Format('%15s: | %5d | %20s | (%d)',
          [inet_ntoa(in_addr(UDPExTable^.Table[I].dwLocalAddr)),
          htons(UDPExTable^.Table[I].dwLocalPort),
          ProcessPIDToName(hProcessSnap, UDPExTable^.Table[I].dwProcessID),
          UDPExTable^.Table[I].dwProcessID]));
    finally
      GlobalFreePtr(UDPExTable);
    end;
  finally
```

```
    // Закрываем хэндл полученый от CreateToolhelp32Snapshot
    CloseHandle(hProcessSnap);
    Memo1.Lines.Delete(0);
    Memo1.Lines.Insert(0,'');
  end; end;
//_____________________выделение пункта MEMO
procedure TForm1.memInfoClick(Sender: TObject);
var   Line: Integer ;
begin
  with (Sender as TMemo) do begin
     Line      := Perform(EM_LINEFROMCHAR, SelStart, 0);
     SelStart  := Perform(EM_LINEINDEX, Line, 0);
     SelLength := Length(Lines[Line]);
  end; end;
procedure TForm1.Memo1Click(Sender: TObject);
var   Line: Integer ;
begin
  with (Sender as TMemo) do begin
     Line      := Perform(EM_LINEFROMCHAR, SelStart, 0);
     SelStart  := Perform(EM_LINEINDEX, Line, 0);
     SelLength := Length(Lines[Line]);
  end; end;
procedure TForm1.AdvStringGrid2DrawCell(Sender: TObject; ACol,
  ARow: Integer; Rect: TRect; State: TGridDrawState);
var s: string;
begin  with Sender as TAdvStringGrid do begin
     s:=cells[acol,arow]; //сохраняем текст из ячейки
     canvas.FillRect(rect);
     DrawText(canvas.handle,pchar(s),-1,Rect, DT_CENTER or DT_WORDBREAK)
end;
end;
procedure TForm1.RadioButton1Click(Sender: TObject);
var
  i,j,i_max,i_min:Integer;
  ar: array [1..7,1..4] of real;
  str : string;
  res1_max,res2_max,res1_min,res2_min:real;
begin   for i := 1 to 7 do
     for j:=1 to 4 do begin
       str:=replace(form1.AdvStringGrid4.Cells[j-1,i],'.',',');
       ar[i,j]:=StrToFloat(str);
     end;
  res1_max:=0; res1_min:=1000;
  i_min:=0; i_max:=0;
  for i := 1 to 7 do begin
     res2_max:=0; res2_min:=0;
     for j:=1 to 4 do begin
        res2_max:=res2_max+ar[i,j];
        res2_min:=res2_min+ar[i,j];
     end;
     if res2_max>=res1_max then begin
       res1_max:=res2_max; i_max:=i;      end;
```

```
    if res2_min<=res1_min then begin
      res1_min:=res2_min; i_min:=i;
    end;   end;
  Label50.Caption:='По критерию Лапласа определены степени защиты: '+#13+#10+
       'наибольшая защита для '+""+AdvStringGrid2.Cells[0,i_max]+""+#13+#10+
       'наименьшая защита для '+""+AdvStringGrid2.Cells[0,i_min]+"";
end;
procedure TForm1.RadioButton2Click(Sender: TObject);
var
  i,j,i_max,i_min:Integer;
  ar: array [1..7,1..4] of real;
  ar1: array [1..7] of real;
  str : string;
  res1_max,res2_max,res1_min,res2_min:real;
begin
  for i := 1 to 7 do
    for j:=1 to 4 do begin
      str:=replace(form1.AdvStringGrid4.Cells[j-1,i],'.',',');
      ar[i,j]:=StrToFloat(str);
    end;
  for i := 1 to 7 do begin
    ar1[i]:=1000;
    for j:=1 to 4 do if ar1[i]>ar[i,j]then ar1[i]:=ar[i,j]
  end;
  i_max:=1; i_min:=1;
  i_min:=0; i_max:=0;
  res1_max:=0; res1_min:=1000;
  for i := 1 to 7 do begin
    if ar1[i]>=res1_max then begin
      res1_max:=ar1[i];
      i_max:=i;
    end;
    if ar1[i]<=res1_min then begin
      res1_min:=ar1[i];
      i_min:=i;
    end;   end;
  if i_max<>i_min then
   Label50.Caption:='По критерию Вальда определены степени защиты: '+#13+#10+
       'наибольшая защита для '+""+AdvStringGrid2.Cells[0,i_max]+""+#13+#10+
       'наименьшая защита для '+""+AdvStringGrid2.Cells[0,i_min]+""
  else Label50.Caption:='Для критерия Вальда должны быть не нулевые все коэффициенты
матрицы';
end;
procedure TForm1.RadioButton3Click(Sender: TObject);
var
  i,j,i_max,i_min:Integer;
  ar: array [1..7,1..4] of real;
  ar1: array [1..7] of real;
  str : string;
  res1_max,res2_max,res1_min,res2_min:real;
begin   for i := 1 to 7 do
    for j:=1 to 4 do begin
```

```
      str:=replace(form1.AdvStringGrid4.Cells[j-1,i],'.',',');
      ar[i,j]:=StrToFloat(str);
    end;
  for i := 1 to 7 do begin  ar1[i]:=0;
    for j:=1 to 4 do if ar1[i]<ar[i,j]then ar1[i]:=ar[i,j]
  end;
  for i := 1 to 7 do  for j:=1 to 4 do ar[i,j]:=ar1[i]-ar[i,j];
  for i := 1 to 7 do begin      ar1[i]:=0;
    for j:=1 to 4 do if ar1[i]<ar[i,j]then ar1[i]:=ar[i,j]
  end;
  i_min:=0; i_max:=0;
  res1_max:=0; res1_min:=1000;
  for i := 1 to 7 do begin
    if ar1[i]>res1_max then begin
      res1_max:=ar1[i];
      i_max:=i;
    end;
    if ar1[i]<res1_min then begin  res1_min:=ar1[i];
      i_min:=i;
    end;
  end;
  Label50.Caption:='По критерию Сэвиджа определены степени защиты: '+#13+#10+
       'наибольшая защита для '+'"'+AdvStringGrid2.Cells[0,i_max]+'"'+#13+#10+
       'наименьшая защита для '+'"'+AdvStringGrid2.Cells[0,i_min]+'"';
end;
procedure TForm1.RadioButton4Click(Sender: TObject);
var   i,j,i_max,i_min:Integer;
  ar: array [1..7,1..4] of real;
  ar1,ar2,ar3: array [1..7] of real;
  str : string;
  res1_max,res2_max,res1_min,res2_min,a:real;
begin   for i := 1 to 7 do  for j:=1 to 4 do begin
str:=replace(form1.AdvStringGrid4.Cells[j-1,i],'.',',');
      ar[i,j]:=StrToFloat(str);
    end;
    //max
  for i := 1 to 7 do begin      ar1[i]:=0;
    for j:=1 to 4 do if ar1[i]<ar[i,j]then ar1[i]:=ar[i,j]
  end;
     //min
  for i := 1 to 7 do begin    ar2[i]:=1000;
    for j:=1 to 4 do if ar2[i]>ar[i,j]then ar2[i]:=ar[i,j]
  end;
  for i := 1 to 7 do ar3[i]:=a*ar2[i]+(1-a)*ar1[i];
  i_min:=0; i_max:=0;
  res1_max:=0; res1_min:=1000;
  for i := 1 to 7 do begin  if ar3[i]>res1_max then begin res1_max:=ar3[i];
      i_max:=i;      end;
    if ar3[i]<res1_min then begin
      res1_min:=ar3[i];
      i_min:=i;
    end;    end;
```

```
  Label50.Caption:='По критерию Гурвица определены степени защиты: '+#13+#10+
       'наибольшая защита для '+""+AdvStringGrid2.Cells[0,i_max]+""+#13+#10+
       'наименьшая защита для '+""+AdvStringGrid2.Cells[0,i_min]+"";
end;
procedure TForm1.AdvStringGrid2SelectCell(Sender: TObject; ACol,
  ARow: Integer; var CanSelect: Boolean);
Begin StatusBar1.Panels[0].Text := '';
  StatusBar1.Panels[1].Text :=arOp[ARow,ACol];
end;
procedure TForm1.AdvStringGrid4SelectCell(Sender: TObject; ACol,
  ARow: Integer; var CanSelect: Boolean);
Begin StatusBar1.Panels[0].Text := '';
  StatusBar1.Panels[1].Text :=arOp[ARow,ACol+1];
end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin   WriteFile();
end;
procedure TForm1.AdvGlowButton1Click(Sender: TObject);
var
  i,j:byte; str:string; res:real;
begin   for i := 1 to 7 do
     for j:=1 to 4 do begin   str:=replace(AdvStringGrid2.Cells[j,i],'.',',');
       res:=StrToFloat(str)*
            StrToFloat(AdvStringGrid1.Cells[1,j-1])*
            StrToFloat(AdvStringGrid3.Cells[1,i-1]);
       AdvStringGrid4.Cells[j-1,i]:=FloatToStr(round(res*1000)/1000);
     end; end;
procedure TForm1.AdvGlowButton2Click(Sender: TObject);
var
  Col, Row: Integer; X, Y : Double; Sg : TAdvStringGrid; S : String; i, j: Integer;
begin   for i:=1 to N do  O[i]:= StrToFloat(AdvStringGrid7.Cells[1,i]);
  for i:=1 to N do
    for j:=1 to N do
      F[i,j]:= StrToFloat(AdvStringGrid8.Cells[i,j]);
  for i:=1 to N do  for j:=1 to N do   OO[i,j]:=O[i];
  for i:=1 to N do begin Res_S[i] := RoundFloat(M(i).S,2);
    Res_Q[i] := RoundFloat(M(i).Q,2);
  end;
  for i:=1 to N do begin
    AdvStringGrid9.Cells[1,i] := Floattostr(Res_S[i]);
    AdvStringGrid9.Cells[2,i] := Floattostr(Res_Q[i]);
  end;
  Sg := AdvStringGrid9;
  Series1.Clear;
  for Row := Sg.FixedCols to Sg.RowCount-1 do begin
    X := StrToFloatDef( Sg.Cells[1, Row ], 0 );
    Y := StrToFloatDef( Sg.Cells[2, Row ], 0 );
    //Сведения для легенды.
    S := '(' + FloatToStr(X) + ', ' + FloatToStr(Y) + ')';
    //Добавляем на график точку. А в легенду добавляем строку S.
    Series1.AddXY(X, Y, S);
    end;  end;  end.
```