Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра телекомунікацій, медійних та інтелектуальних технологій

ДИПЛОМНА РОБОТА МАГІСТРА

Байєсовська мережа і система виявлення зловмисного програмного забезпечення на основі дослідження аномалій

Галузь знань 11 Математика та статистика

Спеціальність 113 Прикладна математика

ДРПМ. 2019/121 01 34

Виконав:
студент 2 курсу, група ПМм-19-1 _____ А. В. Шевцова
Підпис

Керівник:
канд.техн.наук, доцент _____ Т. М. Кисіль
Підпис

До захисту допускаю:
Зав. кафедри ТМІТ д-р.техн.наук, доцент _____ С. К. Підченко
Підпис

3 12 2020 р.

Хмельницький 2020

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

Освітній рівень МАГІСТР

Галузь знань 11 МАТЕМАТИКА ТА СТАТИСТИКА

Спеціальність 113 ПРИКЛАДНА МАТЕМАТИКА

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА ПІДГОТОВКИ МАГІСТРА

ЗАТВЕРДЖУЮ

Зав. кафедри ТМІТ

_____ Підченко С.К.

" 03 " 09 2020 р.

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Шевцовій Анастасії Володимирівні

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Байєсовська мережа і система виявлення зловмисного програмного забезпечення на основі дослідження аномалій

Керівник проекту (роботи) Кисіль Тетяна Миколаївна, к.ф.-м.н, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.09.2020 р. № 118

2. Строк подання студентом проекту (роботи) на кафедру 01.12.2020 р.

3. Вихідні дані до проекту (роботи). Наукові джерела, що стосуються систем виявлення зловмисного програмного забезпечення та Баєсовських мереж

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити).

Аналіз відомих алгоритмів побудови Байєсовської мережі

Аналіз відомих алгоритмів виявлення зловмисного програмного забезпечення

Побудова математичної моделі на базі мереж Байєса

Візуалізація роботи побудованої математичної моделі

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1			
2			
3			

7. Дата видачі завдання «03» вересня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Затвердження теми науковим керівником	01.09.2020 – 02.09.2020	Виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	03.09.2020 –08.09.2020	Виконано
3	Розробка 1 розділу написання ДРМ	09.09.2020 – 20.09.2020	Виконано
4	Аналіз відомих алгоритмів виявлення зловмисного програмного забезечення та побудови мереж Байєса	21.09.2020 – 27.09.2020	Виконано
5	Розробка 2 розділу написання ДРМ	28.09.2020 – 7.10.2020	Виконано
6	Побудова математичної моделі на базі мереж Байєса	08.10.2020 – 13.10.2020	Виконано
7	Розробка 3 розділу написання ДРМ	14.10.2020 – 05.11.2020	Виконано
8	Написання вступу, висновків, формування переліку джерел посилання та додатків	06.11.2020 – 08.11.2020	Виконано
9	Попередній захист дипломної роботи	09.11.2020 –10.11.2020	Виконано
10	Подача роботи на: кафедру, антиплагіат, рецензування, нормоконтроль	12.11.2020 – 3.12.2020	Виконано
11	Захист дипломної роботи	4.12.2020 –15.12.2020	

Студент _____ А. В. Шевцова
Підпис

Керівник проекту (роботи) _____ Т. М. Кисіль
Підпис

АНОТАЦІЯ

Тема дипломної роботи: Байєсовська мережа і система виявлення зловмисного програмного забезпечення на основі дослідження аномалій.

Автор роботи: Шевцова Анастасія Володимирівна.

Керівник роботи: Кисіль Тетяна Миколаївна.

Пояснювальна записка: 71 с., 30 рис., 14 табл., 2 дод., 23 джерел.

Ключові слова: Байєсовська мережа, машинне навчання, принцип мінімальної довжини опису (ОМД), зловмисне програмне забезпечення, аномалії, система виявлення атак, система виявлення зловмисного програмного забезпечення.

Об'єктом дослідження є: Байєсовська мережа та системи виявлення зловмисного програмного забезпечення.

Предметом дослідження є: Можливості використання Байєсовської мережі у системах виявлення зловмисного програмного забезпечення на основі дослідження аномалій в інформаційних системах.

Метою дипломної роботи є: Визначити можливі способи використання Байєсовської мережі для виявлення зловмисного програмного забезпечення на основі дослідження аномалій в інформаційних системах.

Наукова новизна отриманих результатів полягає в тому, що мережі Байєса використовуються для моделювання в біоінформатиці, медицині, класифікації документів, обробці зображень, обробці даних, машинному навчанні і системах підтримки прийняття рішень.

На основі проведених досліджень розроблена математична модель на базі мереж Байєса.

Практична значимість отриманих результатів полягає у можливості вдосконалення методів виявлення зловмисного програмного забезпечення в інформаційних система.

ANNOTATION

Thesis topic: Bayesian network and malware detection system based on the study of anomalies.

Author of the work: Shevtsova Anastasia Vladimirovna.

Scientific adviser: Kisil Tatiana Nikolaevna.

Explanatory note: 71 pages, 30 figures, 14 tables, 2 annex, 23 sources.

Keywords: Bayesian network, machine learning, the principle of minimum description length, malware, anomalies, attack detection system, malware detection system.

The object of research is: Bayesian network and malware detection systems.

The subject of the study is: Possibilities of using the Bayesian network in malware detection systems based on the study of anomalies in information systems.

The purpose of the thesis is: To identify possible ways to use the Bayesian network to detect malicious software based on the study of anomalies in information systems.

The scientific novelty of the results is that Bayesian networks are used for modeling in bioinformatics, medicine, document classification, image processing, data processing, machine learning and decision support systems.

Based on the research, a mathematical model based on Bayesian networks was developed.

The practical significance of the obtained results lies in the possibility of improving the methods of detecting malicious software in information systems.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

СВЗПЗ – система виявлення зловмисного програмного забезпечення

ПЗ – програмне забезпечення

БД - база даних

ІС – інформаційна система

ЗПЗ – зловмисне програмне забезпечення

СВА – система виявлення атак

ОС – операційна система

БМ – Байєсовська мережа

ОМД – опис мінімальної довжини

DDoS - Distributed Denial of Service (розподілена відмова в обслуговуванні)

IDS - система виявлення вторгнень

АПЗ - антивірусне програмне забезпечення

AI – Штучний інтелект

ШНМ - Штучна нейронна мережа

ВСТУП

З розвитком комп’ютерних та інтернет-технологій зросла кількість кібератак. Об’єктом атаки кіберзлочинців можуть стати як фізичні особи, так і великі організації, урядові та військові установи. Злочинці прагнуть отримати доступ до цінної інформації, використовуючи для цього різноманітні мережеві атаки, а також зловмисне програмне забезпечення (ПЗ). Вони отримують доступ до особистих даних через уразливості мереж та інформаційних систем. Досить часто хакери вимагають викуп, що призводить до великих збитків найкрупніших організацій.

На сьогоднішній день більша частина сфер діяльності комп'ютеризована, тому кібератаки представляють величезну загрозу. Разом із технологіями розвивається та вдосконалюється зловмисне ПЗ. В деяких випадках дуже складно розпізнати зловмисне ПЗ через його різноманіття і величезної кількості варіантів зараження.

Регулярно проводиться аналіз зловмисного ПЗ, з метою його вивчення, а також для того, щоб розуміти, як його виявити і усунути. Як правило, подібний аналіз проводиться в ізольованому середовищі, де вивчається його поведінка, можливості, зміст і отримується інформація, яка допоможе запобігти подальшим зараженням. На основі отриманої інформації створюються бази сигнатур вірусів та аномалій, які використовуються в антивірусних програмах та інших системах виявлення зловмисного ПЗ, вторгнень, мережевих атак, також для вдосконалення існуючих і створення нових систем виявлення зловмисного ПЗ. Для цього, досить часто, використовують математичні та ймовірнісні моделі. Байєсовська мережа активно та ефективно використовується у багатьох системах для аналізу даних в сфері бізнесу, медицини, вивчення космічного простору і багатьох інших сферах діяльності людини.

Метою даної роботи є визначення способу використання Байєсовської мережі для виявлення зловмисного ПЗ в інформаційних системах.

Об’єктом дослідження є: Байєсовська мережа та системи виявлення зловмисного програмного забезпечення.

Предметом дослідження є: Можливості використання Байєсовської мережі у системах виявлення зловмисного програмного забезпечення на основі дослідження аномалій в інформаційних системах.

Визначені задачі: розглянути алгоритми для навчання (побудови) Байєсовської мережі, побудувати математичну модель, використовуючи найбільш підходящий алгоритм, візуалізувати її роботу та перевірити правильність та точність розрахунків, використовуючи різні вхідні дані.

Наукова новизна отриманих результатів полягає в тому, що мережі Байєса використовуються для моделювання в біоінформатиці, медицині, класифікації документів, обробці зображень, обробці даних, машинному навчанні і системах підтримки прийняття рішень.

На основі проведених досліджень розроблена математична модель на базі мереж Байєса.

Практична значимість отриманих результатів полягає у можливості вдосконалення методів виявлення зловмисного програмного забезпечення в інформаційних система.

За темою дипломної роботи опублікована одна стаття у фаховому науковому виданні «Збірник наукових праць Конференції АПКН-2020».

1 ЗЛОВМИСНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА АНОМАЛІЇ. МЕРЕЖІ БАЙЄСА

1.1 Зловмисне програмне забезпечення. Мережеві атаки

Зловмисне програмне забезпечення (англ. malware) це будь яке програмне забезпечення, або код, що призначені для несанкціонованого доступу до інформаційної системи, з метою порушення її роботи, викрадення або пошкодження цінної та особистої інформації, виведення з ладу обчислювальної машини. Зловмисне ПЗ може приймати різні форми і маскувати свою присутність в системі. Як правило, таке програмне забезпечення потрапляє в інформаційну систему без згоди власника, через інтернет мережу, з веб-сайтів, що знаходяться під контролем зловмисників, через електронну пошту та прикріплені до неї додатки, а також через USB-накопичувачі.

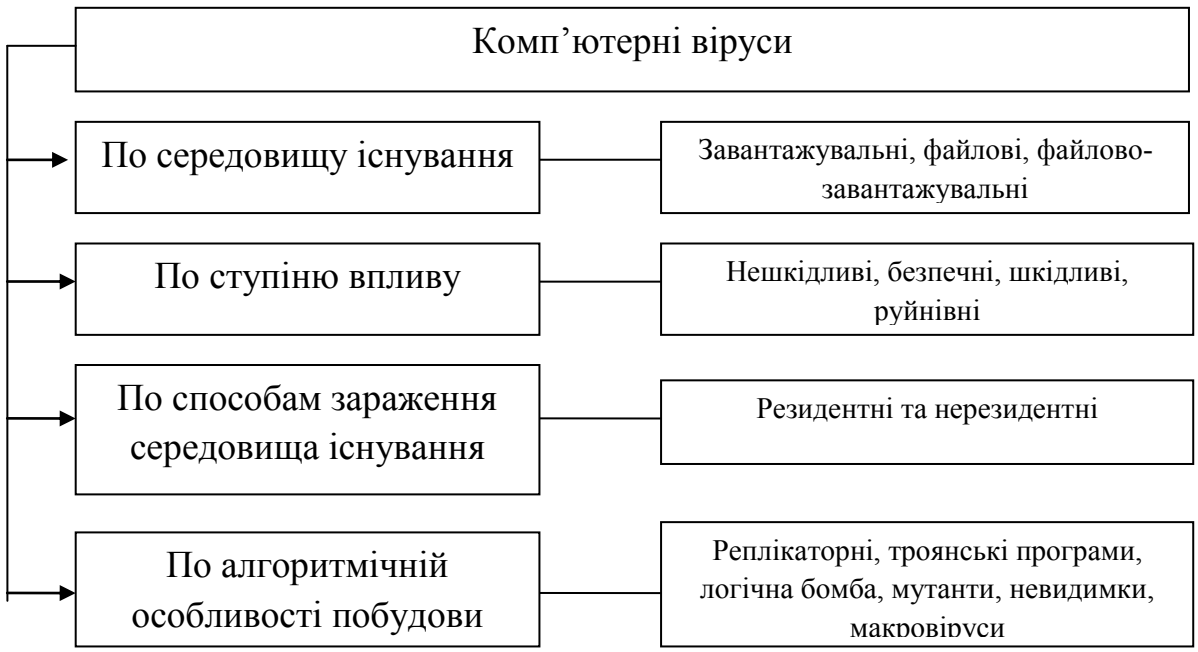


Рисунок 1.1 – Класифікація комп’ютерних вірусів

Уразливість це слабке місце (компонент) в інформаційній системі або корпоративній мережі, використовуючи яке, може бути проведена кібератака.

Причиною виникнення вразливостей в системі можуть виступати деякі фактори, такі як:

— недосконаліть програмного і апаратного забезпечення,

— слабкий рівень захисту інформаційної системи,

— загрози (кібератаки) масової дії,

— непродумана, недопрацьована система зберігання інформації,

— не налаштовані протоколи обміну інформацією в корпоративній мережі або інформаційній системі,

— часткове функціонування інформаційної системи або її системи безпеки.

Вразливості системи можна поділити на 3 типи - об'єктивні, суб'єктивні та випадкові.

Об’єктивні вразливості залежать від апаратного забезпечення інформаційної системи. Суб’єктивні вразливості, це вразливості пов’язані з системою зберігання інформації та порушення інформаційної безпеки системи, такі як: завантаження ПЗ з неперевірених джерел, підключення USB-накопичувачів до локальної корпоративної мережі, налаштування протоколів обміну інформацією некваліфікованим спеціалістом, незаблокований доступ до локальної корпоративної мережі звільнених працівників організації. Випадкові вразливості можуть залежати від багатьох факторів і, як правило, неможливо передбачити їх появу. Факторами появи таких вразливостей можуть виступати несправності в апаратному та програмному забезпеченні інформаційної системи, збої в роботі мережевого обладнання та інші.

Єдиної класифікації зловмисного ПЗ не існує, але його можна класифікувати по функціоналу та направленню атаки:

Вірус - таке зловмисне ПЗ не може бути встановлене без участі користувача та може поширювати свої копії на інші обчислювальні машини, які підключені до загальної мережі.

Червь – ці зловмисні програми здатні поширюватися в комп'ютерній мережі самостійно, так само як і вірус, можуть створювати свої копії.

Троянській конь (Троян, Trojans) – це зловмисне ПЗ, яке маскується під звичайну програму. Не мають власного способу поширення, зазвичай потрапляють в систему за допомогою користувача, який власноруч виконує установку, також можуть бути встановлені іншими зловмисними програмами. Здатні збирати інформацію і пересилати її через мережу.

Троянські програми можна поділити на такі види:

— RAT (Remote Access / Administration Tool), інакше Бекдори (Backdoor, англ. back door - «чорний хід») – це трояни, які здатні надати зловмиснику віддалений доступ до інформаційної системи. Після отримання доступу хакер здатен виконувати дії у зламаній системі: використовувати мікрофон, камеру, встановлювати і видаляти програми та інші. Деякі зловмисні програми здатні фіксувати натискання клавіш на клавіатурі (кейлоггер, keylogger, англ. keyboard logger - клавіатурний регістратор), таким чином можуть надати доступ зловмиснику до логінів і паролів програм та веб-сайтів;

— Вимагачі – відповідно до назви, зловмисники блокують доступ до системи користувачу і вимагають викуп через такі програми, загрожуючи користувачеві різними способами, наприклад, поширенням особистої або конфіденційної інформації користувача в мережі інтернет. Відновити доступ до системи можливо після видалення цієї зловмисної програми;

— Шифрувальники (Virus-Encoder, Trojan-Encoder) – це різновид програм-вимагачів, кріптовірус - використовує криптографію як спосіб зашифрувати файли в системі, тим самим заблокувати доступ

користувачеві. На відміну від звичайних вимагачів, після видалення шифрувальника доступ до системи може лишитися заблокованим;

— Завантажники – ці зловмисні програми призначені для завантаження іншого зловмисного ПЗ або інших файлів в інформаційну систему через інтернет мережу;

— Дезактиватори систем захисту – зловмисні програми, відповідно до назви, видаляють, зупиняють чи блокують системи захисту;

— Банкер – таке програмне забезпечення використовується для виконання кібератак на банківські організації, з метою отримання номера рахунків, pin-коди від карток, та інші дані;

— DDoS-трояни (DDoS – розподілені мережеві атаки, або розподілені атаки типу «відмова в обслуговуванні», англ. Distributed Denial of Service) – зловмисне ПЗ, яке використовується для проведення DDoS атак, для блокування і неможливості використання веб-сайту чи додатка, шляхом відправлення великої кількості запитів на цільову систему, використовуючи обмеження пропускної здатності мережних ресурсів.

Рекламне ПО – зловмисне ПЗ, яке відображає спливаючі вікна з рекламою, рекламні банери, перенаправляє на інші сайти. Зловмисники використовують таке ПЗ, для того, щоб під видом звичайної програми користувач встановив зловмисне ПЗ. Також може бути примусово встановлено з іншими програмними продуктами, як додаткове «корисне» ПО.

Ботнет (англ. Botnet, походить від слів robot і network) або інакше мережа ботів – це комп’ютерна мережа, яка включає в себе комп’ютери, які заражені зловмисним ПЗ (ботами), через які хакери контролюють мережу, та здатні використовувати ці комп’ютери і виконувати будь які дії на них віддалено.

Ботнети можуть мати дві форми:

— модель «клієнт-сервер»,

— однорангову модель (P2P, «Peer-to-peer»).

Поширені типи атак, що здійснюються через ботнет:

— DDoS,

— Спам і моніторинг трафіку,

— Keylogging (використання кейлоггеру),

— Рекламне ПО,

— Клікери (Trojan-Clicker) - збільшення відвідуваності сайтів, накрутка кліків в рекламних мережах.

Руткіт (rootkit, з термінології ОС Linux root – «суперкористувач» і kit – «комплект») – ПЗ, яке здатне приховати сліди присутності кіберзлочинця та роботу зловмисних програм в інформаційній системі. Після встановлення руткітів і отриманні доступу в системі, зловмисник може виконувати дії на зараженій машині, проводити спам- або атаки DDoS. Можуть бути встановлені при підключенні до обчислювальної машини зовнішніх накопичувачів, через заражені веб-сайти та інтернет-посилання.

Руткіти бувають двох видів: виконують дії від імені користувача та ті, що працюють на рівні ядра операційної системи (ОП) – запускаються то старту системи, тим самим повністю її контролюють. Ознакою присутності руткіта в інформаційній системі може стати масова відправка даних по мережі, при неактивних додатках, що можуть використовувати інтернет мережу.

Для створення баз сигнатур вірусів, для вивчення поведінки зловмисного ПЗ, його впливу на інформаційну систему проводиться аналіз за допомогою різних методів. Методи аналізу зловмисного ПЗ можна класифікувати наступним чином:

— Статистичний аналіз – виконується аналіз підозрілого файлу, без його виконання, ідентифікує сигнатуру файлу. Реалізується досить просто. Не здатний виявити всієї інформації, але її може бути достатньо для подальшого аналізу;

— Динамічний аналіз - виконується аналіз підозрілого файлу при його виконанні в ізольованому (віртуальному) середовищі. Реалізується досить просто. Також не здатен виявити всієї інформації, але дозволяє вивчити поведінку зловмисної програми та її вплив на інформаційну систему;
— Аналіз коду - складніший метод аналізу зловмисного ПЗ. Включає в себе і статистичний і динамічний аналіз, виконуючи аналіз шкідливого ПЗ відповідно до можливостей кожного з них. Вимагає знань мови програмування, принципів функціонування операційної системи і її будови;
— Аналіз пам’яті - метод, при якому проводиться аналіз оперативної пам'яті обчислювальної машини з метою збору і аналізу цифрових артефактів. Це допоможе виявити сліди діяльності зловмисного ПЗ, які були приховані і які неможливо виявити іншими доступними методами. Дає можливість зрозуміти, як поводить себе зловмисна програма в зараженому середовищі.

1.2 Дослідження аномалій в інформаційних системах

Існуючі, на сьогоднішній день, методи виявлення зловмисного ПЗ досить специфічні, мають ряд недоліків і не здатні охопити все різноманіття комп'ютерних вірусів.

Найбільш часто використовувані методи:

- Сканування – проводиться перевірка (сканування) файлів та оперативної пам’яті на обчислювальній машині. Сканування виконує пошук зловмисного ПЗ за сигнатурами вірусів, дозволяє виявити ПЗ, яке здатне приховувати (маскувати або шифрувати) свій код. Основні недоліки цього методу: потребує постійного оновлення баз сигнатур вірусів, нездатний виявити зловмисне ПЗ, сигнатура якого відсутня в базі, потребує для своєї роботи постійного використання ресурсів обчислювальної машини;
- Евристичний аналіз – проводить перевірку роботи програм. Відстежуються дії, що виконуються цільовою програмою, з метою фіксування дій, що притаманні зловмисним програмам. Основою цього методу є емпіричні припущення, тому можливі помилкові спрацювання. Основними недоліками цього методу є: метод дуже витратний за часом, можливі помилкові спрацювання, потребує для своєї роботи постійного використання ресурсів обчислювальної машини;
- Виявлення змін – проводиться сканування накопичувачів обчислювальної машини і формуються контрольні суми файлів та важливих областей файлової системи, які порівнюються при кожному скануванні (попередні з поточними). Основні недоліки методу: здатний розпізнати зловмисне ПЗ, яке вже присутнє в інформаційній системі, потребує тільки перевіреного ПЗ на обчислювальній машині, на якій здійснюється пошук вірусів цим методом.

Як правило в антивірусних програмах та інших системах виявлення вторгнень, мережевих атак і зловмисного ПЗ використовують одразу декілька методів. З урахуванням всіх недоліків вже існуючих методів пошуку зловмисного ПЗ є необхідність створення нового методу, який би зміг повністю або частково усунути ці недоліки.

У сучасних системах виявлення атак (англ. - IDS, Intrusion Detection System, СВА) застосовуються механізми, засновані на загальних методах, які в багатьох системах комбінуються.

Класифікувати СВА можна:

• за способом реагування;

• за способом виявлення атаки;

• за способом збору інформації про атаку.

За способом реагування СВА можна так само розділити на пасивні і активні. Пасивні фіксують факт атаки, заносячи дані в журнал і видають попередження. Активні вживають заходи, що б протидіяти атаці.

За способом виявлення атаки СВА прийнято ділити на дві категорії:

• виявлення аномальної поведінки (anomaly-based);

• виявлення зловживань (misuse detection або signature-based).

Технологія виявлення аномальної поведінки вивчає аномальну поведінка користувача (атака, зловмисні дії), тобто поведінку або дію, яка не відповідає нормальній поведінці. Найпростішим прикладом аномальної поведінки може служити велике число з'єднань за короткий проміжок часу, високе завантаження центрального процесора і т. п.

Припустимо, що можливо скласти зліпок нормальної поведінки користувача, тоді будь-яка дія, яка йому не відповідає вважатиметься аномальною. Однак не кожну аномальну дію можна вважати атакою.

У системах з СВА можливі два випадки:

• виявлення аномальної поведінки і віднесення її до класу атак. В даному випадку можливе помилкове спрацьовування;

• пропуск атаки при невідповідності аномальній поведінці. Цей випадок небезпечний тим, що можна пропустити шкідливу атаку.

Технологія виявлення аномальної поведінки покликана виявляти нові типи атак.

Виявлення зловживань полягає в пошуку сигнатури атаки в інформаційній системі. Сигнатурою може виступати, наприклад, шаблон дій, який буде сигналізувати про аномалії. Сигнатури аномалій також, як і сигнатури вірусів зберігаються в БД. Дана технологія виявлення атак схожа на технологію виявлення вірусів. Така система здатна виявити всі відомі види атак, але не здатна виявити нові.

Такі системи часто класифікують за способом збору інформації про атаку:

• виявлення атак на рівні мережі (network-based) - перевіряється мережевий трафік;

• виявлення атак на рівні хоста (host-based) - перевіряються реєстраційні журнали ОС, а також додатки;

• виявлення атак на рівні додатку (application-based) - перевіряються конкретні програми.

На рисунку 1.2 представлена класифікація аномалій в мережі інформаційної системи.

Рисунок 1.2 – Класифікація мережевих аномалій

1.3 Байєсовські мережі. Теорема Байєса.

Теорема Байєса (формула Байєса, формула гіпотез) — одна з теорем теорії ймовірностей. Дозволяє розрахувати ймовірність події А, за умови, що відбулася інша взаємозалежна з нею подія В. Формула Байєса має такий вид:

$$P(A|B) = P(B|A)\,P(A)/P(B),$$

де

P (A | B) - ймовірність настання події А, за умови, що відбулася подія В (апостеріорна ймовірність або умовна ймовірність);

P (B | A) - ймовірність настання події В, за умови, що відбулася подія А (апостеріорна ймовірність);

P (A) - повна ймовірність настання події А (апріорна ймовірність);

P (B) - повна ймовірність настання події В (апріорна ймовірність).

Розглянемо приклад:

До проведення експерименту про його умови можна було висунути ряд несумісних припущень (гіпотез), які утворюють повну групу: $B_1, B_2, \ldots, B_n$.

Ймовірності гіпотез є апріорними і рівні:

$$P\ (A|B_1), P\ (A|B_2), \ldots, P\ (A|B_n).$$

Після проведення експерименту відбулась подія А.

Використовуючи формулу Байеса можливо скорегувати апріорні ймовірності гіпотез для визначення причин настання події *A*, за умови того, що подія вже відбулася.

Байєсова мережа (або Байєсова мережа довіри) - це імовірнісна модель, що складається з множини змінних і їх імовірнісних залежностей. Являє собою спрямований ациклічний граф, вершинами (вузлами) якого є змінні, а ребрами - умовні залежності між цими змінними. При проході ациклічного графа з будь-якого вузла, буде пройдений не весь граф, а тільки його частина.

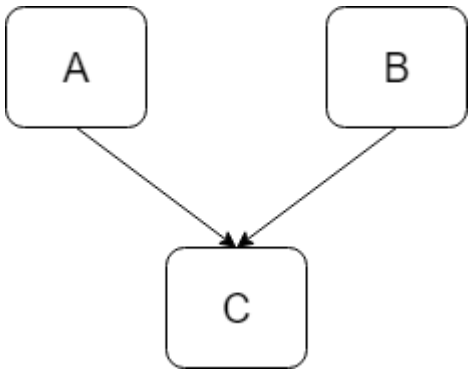


Рисунок 1.3 - Найпростіша Баєсова мережа

Вершини А і В є батьківськими по відношенню до вершини С, а вершина С – їх нащадком. Для вершин, в яких є батьківські вершини формуються таблиці умовних ймовірностей, вони демонструють стани, в яких може перебувати вузол, в залежності від станів вершин, від яких вона є

залежною. Ймовірності вершин які не мають батьківських вузлів не від чого не залежать, тобто є маргінальними.

Вузол С залежить від вузлів А і В, тому стани, в яких може перебувати вершина С, так само залежать від станів вершин А і В. Відповідно до цього, ймовірність перебування вузла С в різних станах можна записати формулою:

$$P\ (C_k) = \sum_i \sum_j P(C_k|A_i\ , B_j)P(A_i, B_j)\ .$$

Вершини А і В є незалежними подіями, тому спільна ймовірність цих подій дорівнює:

$$P\ (A_i, B_j)\ =\ P\ (A_i)\ P\ (B_j).$$

Зв'язки між вузлами в БМ бувають декількох видів:

— послідовний (рисунок 1.4);

— збіжний (рисунок 1.5);

— розбіжний (рисунок 1.6).

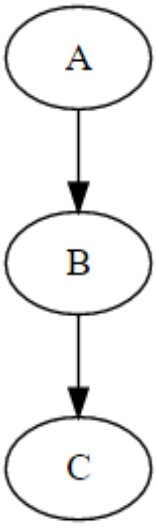


Рисунок 1.4 – Послідовний зв'язок вузлів БС

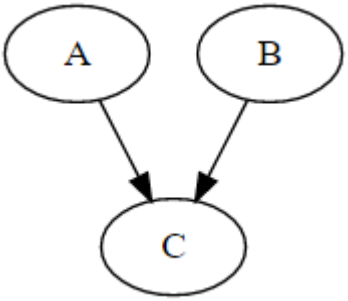


Рисунок 1.5 – Збіжний зв'язок вузлів БС

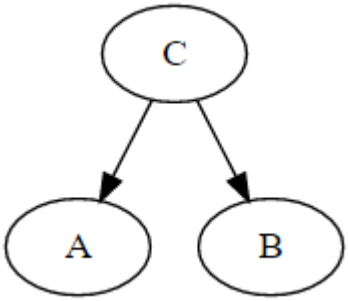


Рисунок 1.6 – Розбіжний зв'язок вузлів БС

1.4 Висновки

В даному розділі дипломної роботи було розглянуто такі поняття як теорема Байєса, мережі Байєса, класифікація та види зловмисного програмного забезпечення, типи аналізу зловмисного програмного забезпечення, мережеві атаки та способи їх виявлення.

2 НАВЧАННЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ

2.1 Машинне навчання

Машинне навчання (Machine learning, ML) - це набір методів і алгоритмів в області штучного інтелекту, які застосовують для навчання систем, використовуючи для цього величезні масиви даних, серед яких системі необхідно знаходити закономірності.

Штучний інтелект (Artificial intelligence, AI) - це сукупність різних технологічних і наукових рішень, методів і алгоритмів, які дозволяють створювати програми за подобою інтелекту людини.

Цілі AI:

- Логічне міркування. Пристосувати комп'ютери для виконання складних завдання, з якими люди не здатні впоратися самостійно;
- Подання знань. Дозволити комп'ютерам описувати об'єкти, людей і мови. Для прикладу можна навести об'єктно-орієнтовану мову програмування Smalltalk;
- Планування і навігація. Дати можливість комп'ютерам добиратися з пункту А в пункт Б. Наприклад, перший самоврядний робот був побудований на початку 1960-х;
- Обробка природної мови. Адаптувати комп'ютер до розуміння і обробки мови. Наприклад здійснювати переклад з однієї мови на іншу;
- Сприйняття. Пристосувати комп'ютер взаємодіяти зі світом через зір, слух, дотик і запах;
- Емпіричний інтелект. Інтелект, який не запрограмований явно, а формується поступово з інших цілком штучно-інтелектуальних особливостей. Метою є комп'ютери, які можуть симулювати емоційний інтелект, моральне судження і інші властивості.

Області застосування штучного інтелекту

- Машинне навчання;
- Пошук і оптимізація - сюди можна віднести такі алгоритми, як алгоритм найшвидшого спуск;
- Задоволення обмежень - це процес пошуку рішень при встановлених обмеженнях;
- Логічне міркування. Прикладом може служити експертна комп'ютерна система, що моделює здатність приймати рішення як людина;
- Розподіл усіх міркування - це поєднання теорії ймовірності з дедукційною логікою;
- Теорія контролю (управління) - зазвичай це система диференціальних рівнянь, які описують фізичну систему, на зразок робота або авіаційного транспорту.

Data science - це наука про дані, методи аналізу даних, які дозволяють отримати з даних максимум корисної інформації. Вона тісно пов'язана з машинним навчанням, наукою про мисленні (Cognitive Science), а також з технологіями для роботи з великими даними (Big Data).

Розробка алгоритмів машинного навчання стала можливою після появи штучного інтелекту. Першою програмою, здатною самостійно навчатися, прийнято вважати гру в шашки, яка була винайдена Артуром Самуелем в 1952 році. Програма була здатна аналізувати поточні позиції і вибирати найкращі варіанти для подальших ходів. У той час визначення машинного навчання звучало, як область досліджень розробки машин, які не є заздалегідь запрограмованими. Пізніше, в 1957 році була запропонована модель нейронної мережі, яка схожа на сучасні алгоритми машинного навчання. З того часу ведеться розробка моделей і систем машинного навчання: алгоритм дерева рішень (1986 рік), метод опорних векторів (1995 рік), глибоке навчання (Deep Learning - з 2005 року) і т.д.

Метою машинного навчання є видавати максимально точні прогнози, використовуючи вхідні дані. Це важливо, тому що на основі цих прогнозів, наприклад маркетологи або співробітники, будуть приймати рішення. На сьогоднішній день машинне навчання застосовується для великої кількості програм з різних сфер діяльності людини: банки, ресторани, роботи на виробництві і т.д.

Основою для машинного навчання є теорія ймовірності і математична статистика. Багато алгоритми машинного навчання являють собою логічне продовження процедур статистичного моделювання.

Переваги машинного навчання:

— Інтелектуальне управління великими даними;

— Інтелектуальні пристрої - сюди можна віднести пристрої від персональних моніторів здоров'я і фітнес-трекерів до самоврядних автомобілів;

— Великі можливості для споживачів - машинне навчання дозволяє за допомогою механізмів пошуку, веб-додатків і інших технологій коригувати результати і рекомендації відповідно до своїх уподобань користувачів, що дозволяє вивести персональне обслуговування споживача на неймовірно високий рівень.

Для успішного застосування машинного навчання необхідно відповісти на наступні питання:

— Що необхідно спрогнозувати?

— Які вхідні дані оптимально використовувати для цього?

— Чи відповідає результат очікуванням?

— Чи є винятки, які потрібно врахувати?

— Як застосовувати отримані результати?

Рисунок 2.1 - Класифікація постановок задач машинного навчання

Можна виділити два основні класи задач машинного навчання:

— завдання навчання з учителем (supervised learning) - в даному випадку на вхід подається набір навчальних прикладів (training set, training sample, тренувальна вибірка). Завдання в тому, що необхідно продовжити вже відомі відповіді на новий досвід, який подається у вигляді тестового набору даних (test set, test sample). Дані, що використовуються для навчання моделі, повинні бути схожі з тими даними, на яких буде застосована дана модель. Завдання навчання з учителем діляться на завдання класифікації і регресії. У першому випадку, що подається на вхід об'єкт необхідно віднести до одного з класів. У другому випадку, необхідно передбачити значення (їх може бути безліч), деякої функції, наприклад розрахувати вагу людини по його зростанню. Потім ці значення подаються на вхід класифікатором.

— завдання навчання без вчителя (unsupervised learning) - такі завдання використовують в тому випадку, якщо немає необхідного набору даних для конкретного завдання.

2.2 Алгоритми навчання математичних моделей

Популярні алгоритми машинного навчання:

- Наївний Байєсовський класифікатор (навчання з вчителем - класифікація) - спирається на теорему Байеса і дозволяє передбачати клас на основі заданого набору ознак, використовуючи ймовірності;
- Алгоритм k-середніх (навчання без вчителя - кластеризація) - використовується для класифікації даних, які не розбиті на класи. Проводиться пошук груп в рамках даних, число груп представлено змінною k і до однієї з груп присвоюється одиниця даних ітеративно;
- Метод опорних векторів (навчання з вчителем - класифікація) - аналізує дані, які використовуються для класифікації та регресійного аналізу, сортують дані по категоріях;
- Лінійна регресія (навчання з вчителем - регресія) - дозволяє зрозуміти взаємозв'язок між двома безперервними змінними;
- Логістична регресія (навчання з вчителем - класифікація) - проводить оцінку ймовірності виникнення події на основі доступних даних з минулого;
- Штучна нейронна мережа (навчання з підкріпленням) - ґрунтуються на таких біологічних системах як мозок і використовують той же принцип обробки інформації;
- Дерево рішень (навчання з вчителем - класифікація / регресія) - це деревоподібна структура, яка використовує метод розгалуження для відображення можливих результатів прийняття рішень. Вузол дерева відповідає перевірці умови по конкретній змінній, а гілка це результат цієї перевірки;

— Випадковий ліс (навчання з вчителем - класифікація / регресія) - це метод об'єднує безліч алгоритмів і показує кращі результати в класифікації, регресії та інших завданнях;

— Метод k-найближчих сусідів (навчання з вчителем) - оцінює ймовірність належності даних до однієї з груп. Переглядає сусідні точки даних від обраної і визначає до якого класу ця точка належить;

— Метод найшвидшого спуск і лінія максимальної відповідності умовам - дозволяє зробити точний прогноз на основі отриманих даних.

EM-алгоритм (англ. Expectation-maximization) - алгоритм, який використовується для знаходження оцінок максимальної правдоподібності параметрів імовірнісних математичних моделей, в тому випадку, коли модель залежить від деяких прихованих змінних. Кожна ітерація алгоритму складається з кроку E і M. На E-кроці (expectation) обчислюється очікуване значення функції правдоподібності. На M-кроці (maximization) обчислюється оцінка максимального правдоподібності, це значення буде використано в кроці E в наступній ітерації. Алгоритм виконується до збіжності.

EM-алгоритм це метод машинного навчання без учителя, в разі коли заздалегідь не відомі істинні відповіді.

EM-алгоритм застосовується для вирішення завдань двох типів:

— завдання, пов'язані з аналізом неповних даних, коли деякі статистичні дані відсутні,

— завдання, в яких функція правдоподібності має вигляд, який не допускає зручних аналітичних методів дослідження, але допускає спрощення, якщо в задачу ввести додаткові приховані змінні (завдання розпізнавання образів, реконструкції зображень, завдання кластерного аналізу).

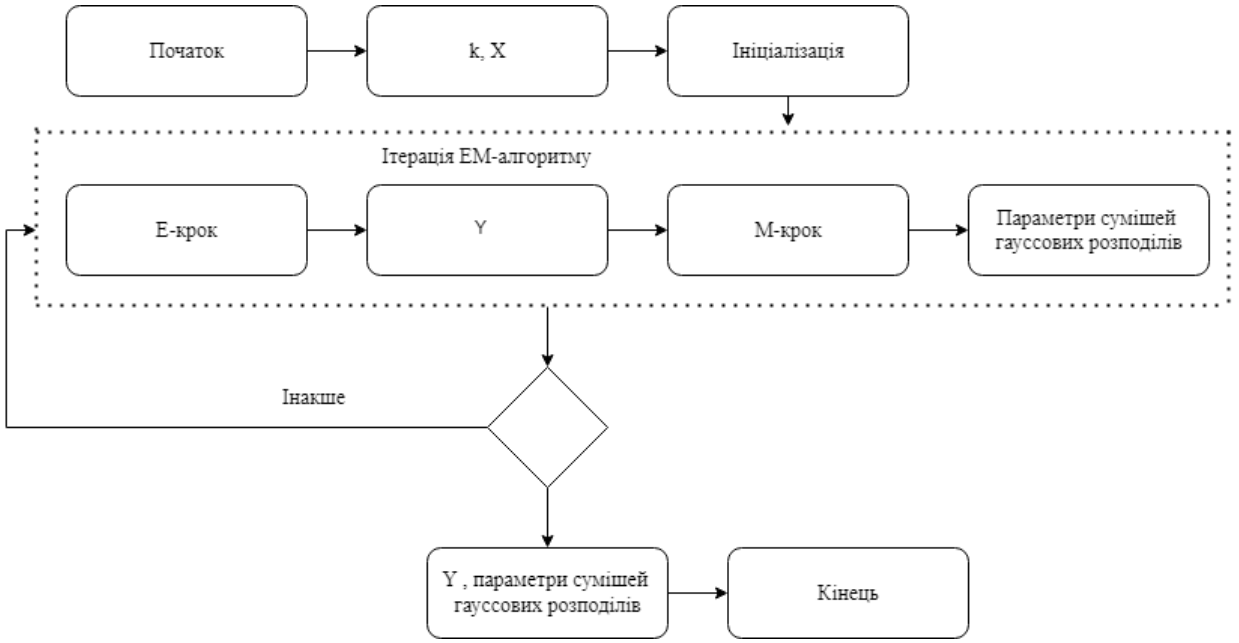


Рисунок 2.2 – Граф ЕМ-алгоритму

Наївний Байєсовський алгоритм

Даний алгоритм перевершує іншими алгоритмами класифікації за швидкістю роботи з даними. Він ґрунтується на використанні теореми Байєса. Алгоритм допускає наявність незалежності ознак, тобто, наявність якої-небудь ознаки не пов'язане з наявністю іншої якої-небудь ознаки. І навіть якщо одна ознака залежить від іншої або кількох ознак, то кожна з них все одно вносить незалежний внесок у ймовірність. Саме тому алгоритм називають «наївним». Цей алгоритм дуже добре справляється з великими обсягами даних. З 1950-х років активно використовувався і вивчався аналіз з використанням Наївного Байєсовського класифікатора в області класифікації документів. Даний алгоритм масштабується по числу ознак. Даний класифікатор, як і будь-який інший, привласнює мітки класів спостереженням, представленим векторами ознак. Простий Байєсовський класифікатор будується на основі навчання з учителем.

Типи наївного Байєсовського класифікатора:

— Поліноміальний Наївний Байєс. Використовується для класифікації документів за категоріями (спорт, політика і т.д.). Використовувані

класифікатором ознаки - це частота слів, що зустрічаються в документі;

— Бернуллі Наївний Байєс. Ознаками є булеві змінні. Параметри, що використовуються для прогнозування змінної класу, приймають тільки два значення - так чи ні;

— Гаусовський Наївний Байєс. Коли ознаки приймають безперервне значення і не є дискретними, передбачається, що ці значення взяті з Гаусовського розподілу (рисунок 2.3).

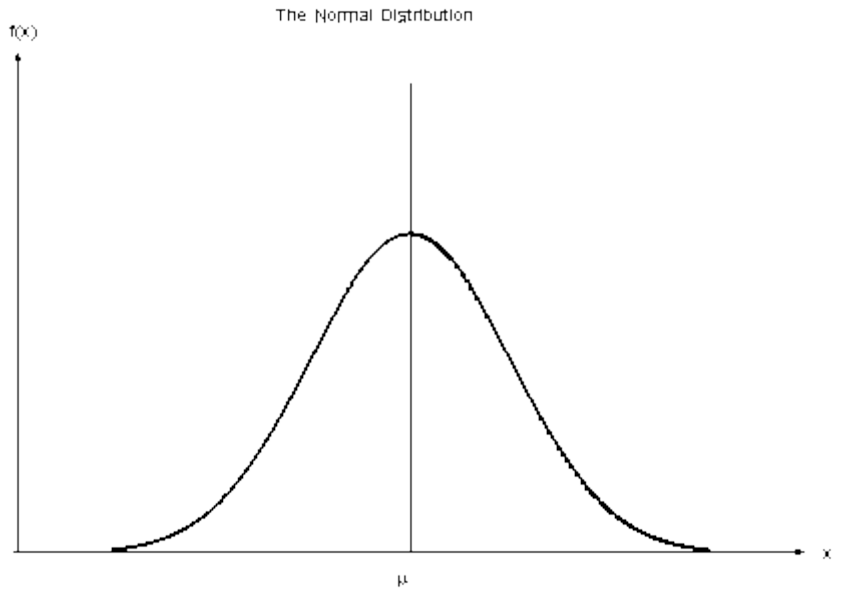


Рисунок 2.3 - Гаусовський розподіл (нормальний розподіл)

Переваги наївного Байєсовського алгоритму:

— Класифікація об'єктів виконується легко і швидко;
— Коли допущення про незалежність ознак виконується, Наївний Байєсовський алгоритм перевершує інші алгоритми і вимагає менший обсяг навчальних даних;
— Наївний Байєсовський алгоритм краще працює з категорійними ознаками, ніж з безперервними. Для безперервних ознак передбачається нормальний розподіл, що є досить сильним допущенням.

Недоліки наївного Байєсовського алгоритму:

— Якщо в наборі даних є певне значення категорійної ознаки, яке було відсутнє в навчальному наборі даних, тоді модель присвоїть нульову ймовірність для цього значення і не зможе зробити прогноз. Це явище відоме під назвою «нульова частота» (zero frequency). Дана проблема вирішується за допомогою методів згладжування, наприклад згладжування по Лапласу (Laplace smoothing);
— Прогнози не завжди можуть бути достатньо точними;
— Набори повністю незалежних ознак зустрічаються дуже не часто.

Розглянемо Метод найшвидшого спуску і лінії максимальної відповідності умовам. Нехай є дані про зріст і вагу деяких людей. Необхідно вгадати вагу людини по її зросту. Приблизно правильну відповідь можна дати при наявності графіка тренда даних. Цей графік-лінія показував би очікуваний вага для кожного значення зростання. В такому випадку необхідно знайти значення зросту, рухаючись паралельно осі ваг і дійшовши до лінії тренда можна буде дізнатися вагу. Щоб побудувати такий графік власноруч знадобиться занадто багато часу. У таких випадках застосовують Метод найшвидшого спуску. На рисунку 2.4 наведений графік, побудований за допомогою даного методу.

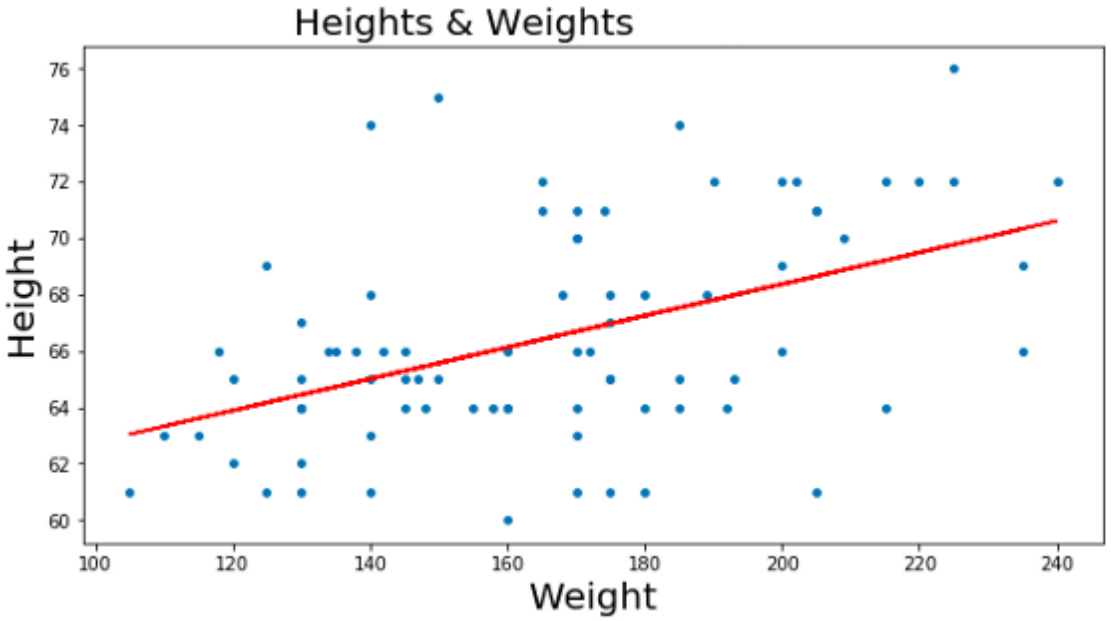


Рисунок 2.4 – Лінія максимальної відповідності умовам

Алгоритм Випадковий ліс (Random Forest) був придуманий Лео Брейманом і Адель Катлер.

Random Forest - це безліч вирішальних дерев.

Роботу алгоритму Random Forest можна розбити на декілька кроків:

— Вибір випадкових вибірок з заданого набору даних;

— Побудувати дерево рішень для кожної вибірки. Отримати результат прогнозування з кожного дерева;

— Провести голосування для кожного прогнозованого результату;

— Вибрати результат прогнозу з найбільшою кількістю голосів в якості остаточного результату прогнозу.

На рисунку 2.5 представлена діаграма, що ілюструє роботу алгоритму Random Forest.

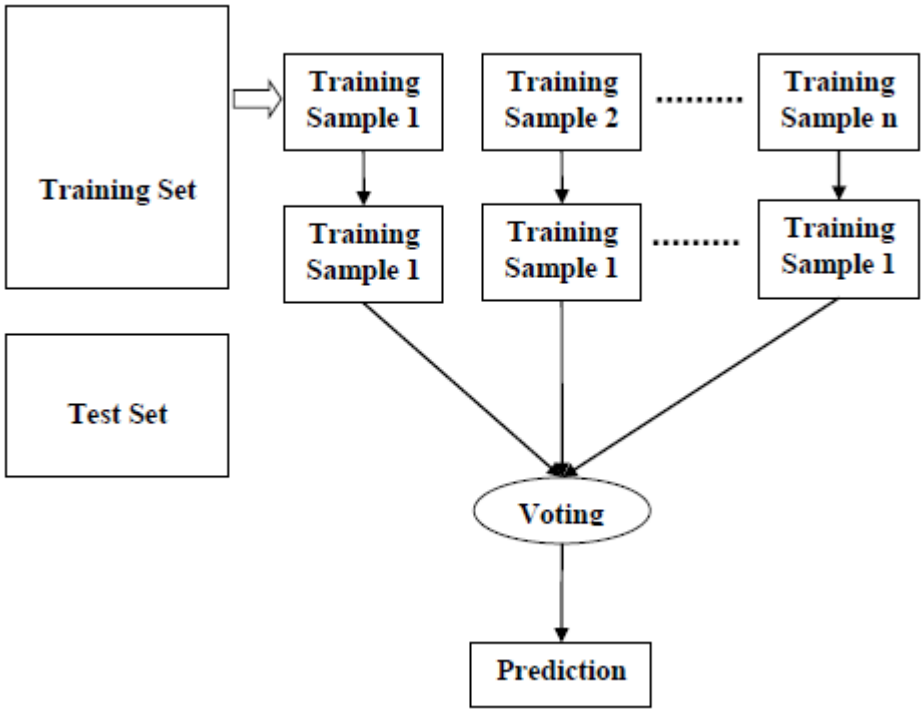


Рисунок 2.5 – Робота алгоритму Random Forest

Переваги алгоритму Random Forest:

— Вирішує проблему переоснащення допомогою усереднення або об'єднання результатів різних дерев;

— Працює краще з великою кількістю даних, ніж одне дерево;

— Має меншу дисперсію, ніж одне дерево;

— Висока точність;

— Зберігає хорошу точність після надання даних без масштабування;

— Підтримує високу точність при відсутності частини даних.

Недоліки алгоритму Random Forest:

— Основним недоліком є складність алгоритму;

— Забирає більше часу для побудови, ніж дерева рішень;

— Потрібно більше обчислювальних ресурсів для реалізації;

— Трудомісткий процес прогнозування в порівнянні з іншими алгоритмами.

Дерево прийняття рішень це засіб підтримки прийняття рішень при прогнозуванні, що широко застосовується в статистиці і аналізі даних.

Дерево рішень, як і звичайне дерево, має гілки і листя. Гілки, або ж ребра графа, зберігають значення атрибутів, а листя - значення цільової функції. Існують так само батьківські і дочірні вузли.

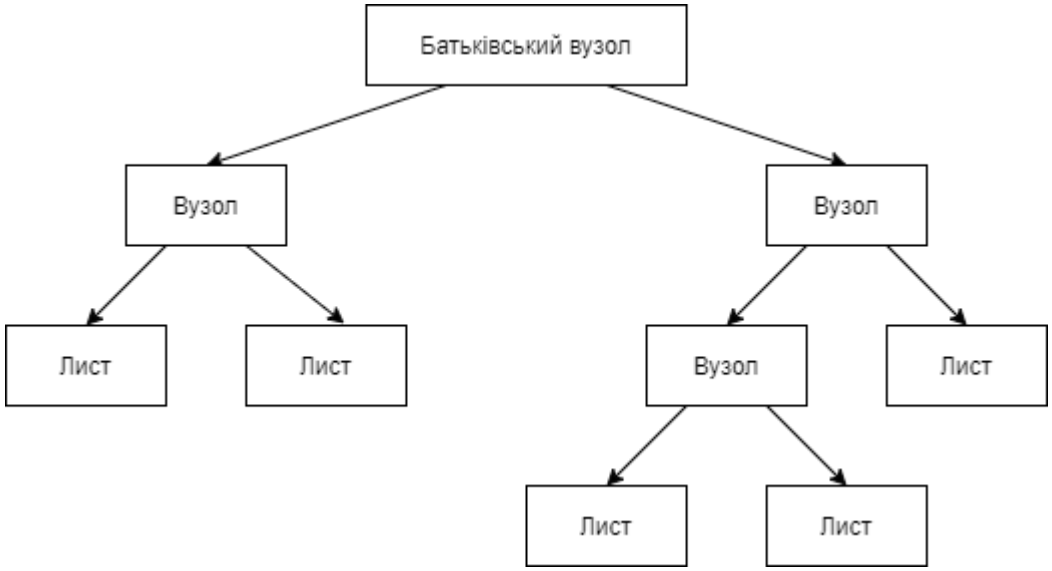


Рисунок 2.6 – Будова Дерева рішень

Загальний алгоритм побудови Дерева прийняття рішення складається з трьох кроків:

— Вибір атрибуту A і розміщення його у батьківський (кореневий) вузол;

— З набору даних для кожного значення атрибуту вибираються тільки ті, для яких A=i;

— Рекурсивно будується дерево рішень.

Існує декілька алгоритмів прийняття рішень за допомогою яких можна визначити атрибут A:

— ID3: Щоб визначити атрибут, необхідно підрахувати ентропію всіх невикористаних ознак і вибрати ту, для якої ентропія мінімальна.

Цей атрибут і буде вважатися найбільш доцільною ознакою класифікації,

— C4.5: Є удосконаленням методу ID3. Присутня можливість розбиття області значень незалежної числової змінної на кілька інтервалів, кожен з яких буде атрибутом,

— CART: Алгоритм розроблений для побудови бінарних дерев рішень (кожен вузол дерева має тільки двох нащадків). Алгоритм працює за принципом поділу множини прикладів на дві рівні частини. Правий нащадок відображає ті приклади, в яких правило виконується, лівий нащадок - ті, де правило не виконується.

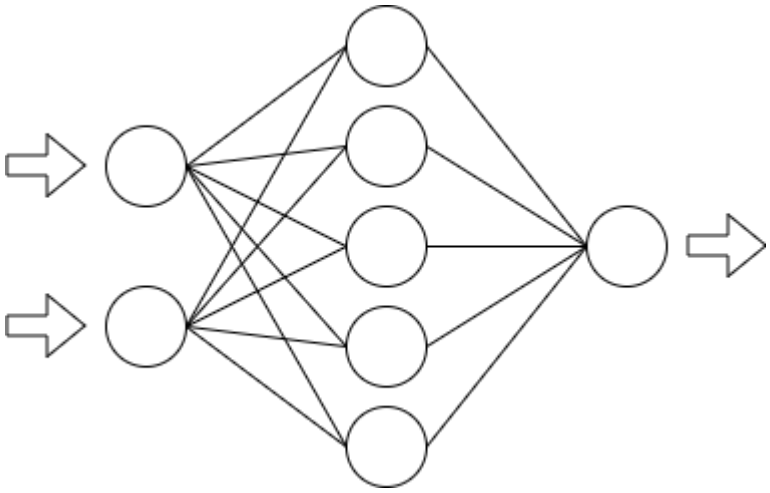


Рисунок 2.7 – Найпростіша нейронна мережа

Штучна нейронна мережа (ШНМ) - це математична модель, а також її програмна і апаратна реалізація. Побудована за принципом функціонування біологічних нейронних мереж живого організму.

Штучна нейронна мережа складається з штучних нейронів (artificial neuron), що представляє собою спрощену модель біологічного нейрона. Штучний нейрон приймає сигнали з багатьох входів, обробляє їх єдиним способом і передає результат на багатьом іншим штучним нейронам. Зв'язки між штучними нейронами називаються синапсами. У синапсу є один параметр - ваговий коефіцієнт. Значення вагового коефіцієнта впливає на

зміну інформації, при її передачі від одного нейрона до іншого. Саме підбір вагового коефіцієнта для кожного синапсу призводить до отримання необхідного результату. На рисунку 2.7 зображена найпростіша нейронна мережа.

Класифікація нейронних мереж:

— По типу вхідних даних:

- Аналогові (дійсні числа),
- Двійкові (двійкові числа),
- Образні (знаки, ієрогліфи, символи),

— По характеру навчання:

- З вчителем,
- Без вчителя
- З підкріпленням,

— По характеру налаштування синапсів:

- З фіксованими зв’язками,
- З динамічними зв’язками (налаштування синаптичних зв'язків в процесі навчання),

— По часу передачі сигналів:

- Синхронні мережі,
- Асинхронні мережі,

— По характеру зв’язків:

- Мережі прямого поширення,
- Рекурентні мережі,
- Рекурентна мережа Хопфілда,
- Двонаправлені мережі.

Задачі, які можуть бути вирішені за допомогою ШНМ:

— Розпізнавання образів: На сьогоднішній день в цій галузі нейронні мережі застосовуються найчастіше. Образами можуть виступати символи, зображення, звуки і т.д.;

— Класифікація: Розподіл даних по параметрам;
— Прийняття рішень і управління: Дуже схоже на класифікацію, тільки в якості даних для розподілу виступають події;
— Кластеризація: Розподіл даних по категоріям, що заздалегідь не відомі;
— Прогнозування: Здатність передбачити майбутні значення деякої послідовності на основі попередніх значень і факторів у теперішньому часі;
— Апроксимація;
— Стиснення даних і асоціативна пам'ять: Здатність знаходити взаємозв’язок між параметрами, що дозволяє представити дані компактніше. Асоціативна пам’ять відповідає за зворотній процес – процес відновлення даних.

Переваги ШНМ:

— Рішення задач в умовах невизначеності. Можливість працювати з неповними даними;
— Стійкість до шумів у вхідних даних. Нейронна мережа здатна самостійно виявляти неінформативні для аналізу параметри і відсіяти їх, тому немає необхідності в попередньому аналізі вхідних даних;
— Гнучкість структури нейронних мереж. Нейрони і їх зв'язки можна комбінувати різними способами. Це дає можливість одному нейрокомп'ютеру вирішувати різні завдання;
— Висока швидкодія. Вхідні дані обробляються відразу багатьма нейронами одночасно;
— Адаптація до змін навколишнього середовища. Здатні підлаштовуватися під мінливі умови навчаючись;
— Відмовостійкість нейронних мереж. На несприятливі зміни умов Нейронна мережа реагує незначним зниженням продуктивності.

Недоліки ШНМ:

— Відповідь завжди приблизна. Нейронні мережі не здатні давати точні і однозначні відповіді;
— Нездатність прийняття рішень в кілька етапів. Нейронна мережа здатна вирішувати завдання тільки "в один захід";
— Нездатність вирішувати обчислювальні завдання. Нейронна мережа не може, наприклад вирішити рівняння;
— Трудомісткість і тривалість навчання. Для навчання Нейронної мережі необхідно використовувати величезні обсяги даних.

Алгоритм k-найближчих сусідів (K-nearest neighbor)

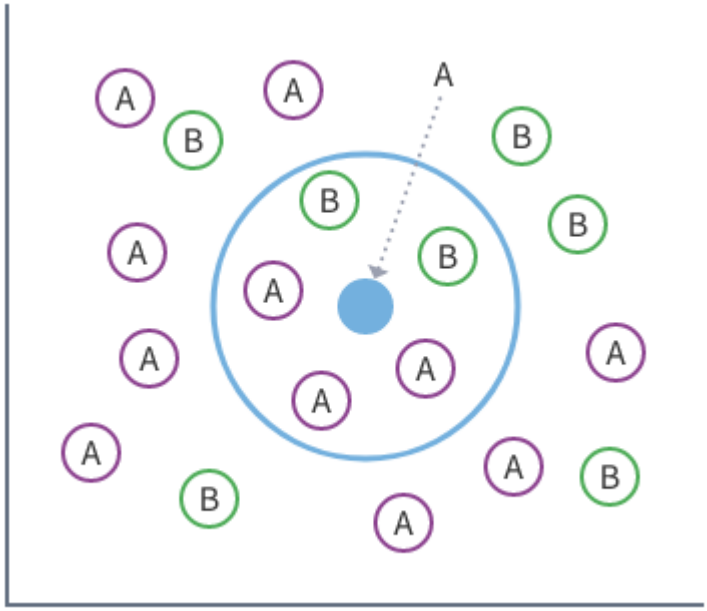


Рисунок 2.8 - Алгоритм k-найближчих сусідів

Основна ідея алгоритму в тому, що для точки, для якої необхідно зробити прогноз, можна використовувати точки, що лежать поруч, цільові змінні яких вже відомі. Це зображено на рисунку 2.8. Є ряд навчальних даних, які мають два типи цільових змінних - це двійкова класифікація. Інакше цільові змінні можна назвати мітками або класами.

Переваги Алгоритму k-найближчих сусідів:

— Стійкий до аномальних викидів;

— Нескладна програмна реалізація алгоритму;

— Результат роботи алгоритму легко піддаються інтерпретації;

— Можливість модифікації алгоритму, що дозволяє можливість налаштувати алгоритм під конкретну задачу.

Недоліки Алгоритму k-найближчих сусідів:

— Набір даних, який використовується для алгоритму, повинен бути репрезентативним;

— Модель не можна "відокремити" від даних. Для класифікації нового прикладу потрібно використовувати всі приклади. Ця особливість сильно обмежує використання алгоритму.

2.3 Використання алгоритмів для навчання мереж Байєса

Наївний Байєсовський алгоритм

В основі Байєсовської класифікації лежить гіпотеза максимальної ймовірності, тобто вважається, що об'єкт d належить класу $c_j (c_j \in C)$, якщо при цьому досягається найбільша апостеріорна ймовірність: $\max_c P(c_j|d)$. По формулі Байєса,

$$P(c_j|d) = \frac{P(c_j)P(d|C_j)}{P(d)} \approx P(c_j)P(d|c_j),$$

де

$P(d|c_j)$ це ймовірність зустріти об’єкт d серед об’єктів класу c_j. $P(c_j)$ і $P(d)$ це апріорні ймовірності класу c_j та об’єкту d (остання, не впливає на вибір класу і може бути опущена).

Якщо зробити "наївне" припущення, що всі ознаки, що описують об'єкти класифікації, абсолютно рівноправні між собою і не пов'язані один з одним, то $P(d|c_j)$ можна обчислити як добуток ймовірностей ознак, що зустрічаються $x_i (x_i \in X)$ серед об'єктів класу c_j:

$$P(d|c_j) = \prod_{i=1}^{|X|} P(x_i|c_j),$$

де

$P(x_i|c_j)$ це ймовірнісна оцінка внеску ознаки x_i в те, що $d \in c_j$.

На практиці при множенні дуже малих умовних ймовірностей може спостерігатися втрата значущих розрядів, в зв'язку з чим замість самих оцінок ймовірностей $P(x_i|c_j)$ застосовують логарифми цих ймовірностей. Оскільки логарифм це монотонно зростаюча функція, то клас c_j з найбільшим значенням логарифма ймовірності залишиться найбільш імовірним. Тоді вирішальне правило наївного Байєсовського класифікатора (Naive Bayes Classifier) приймає наступний остаточний вигляд:

$$c^* = arg_{c_j \in C} \max [\log P(c_j) + \sum_{i=1}^{X} P(x_i|c_j)].$$

Далі необхідно подбати, щоб значення логарифмуємих ймовірностей були не надто близькі до 0, для цього можна застосувати Лапласовське згладжування.

Бернуллі Наївний Байєс (модель Бернуллі)

Нехай *x* - об'єкт з вибірки, x^k - його k-та ознака.

$w_k, k = 1,2,\ldots,D$ - всі унікальні слова в корпусі.

$$x^k = \text{II}\ [w_k \in x],$$
$$\theta_y^k = p(x^k = 1|y).$$

Висновок вирішального правила:

$$(1)\hat{y}(x) = \arg\max_y p(y|x) = \arg\max_y p(y)p(x|y) =$$
$$(2) = \arg\max_y p(y)\prod_{k=1}^{D} p(x^k|y) = \arg\max_y p(y)\prod_{k=1}^{D}\left(\theta_y^k\right)^{x^k}(1-\theta_y^k)^{1-x^k} =$$
$$= \arg\max_y ln\ p(y) + \sum_{k=1}^{D}(x^k \ln\theta_y^k + (1-x^k)\ln(1-\theta_y^k)),$$

де

$p(y) = \frac{N_y}{N}, \theta_y^k = \frac{N_{yk}+\alpha}{N_y+2\alpha}$ - емпіричні оцінки ймовірностей,

N_y – кількість документів класу y,

N_{yk} - кількість документів класу y, що містять w_k,

(1) - Байєсовське правило максимальної апостеріорної ймовірності класів,

(2) - припущення "наївного Байєса".

Гаусовський Наївний Байєс (Гаусовський класифікатор)

Основна ідея в тому, що б побудувати класифікатор в припущенні того, що функція $p\,(x \mid y)$ (так звана функція правдоподібності, тобто розподіл об'єктів при фіксованій відповіді y) відома для кожного класу і дорівнює щільності багатовимірного нормального (Гаусовського) розподілу:

$$p(x|y) = N(\mu_y, \sum_y) = \frac{1}{\sqrt{(2\pi)^D|\det(\sum_y)}}\exp\left(-\frac{1}{2}(x-\mu_y)^T\sum_y^{-1}(x-\mu_y)\right),$$
$$y \in \{1,2,\dots,C\},$$

де

$\sum_y$- матриця коваріації,

μ_y - вектор математичних очікувань,

N – кількість об’єктів,

D – розмірність ознакового простору.

Таким чином, параметрами Гаусовського класифікатора є апріорні розподілення $p\,(y)$, вектори математичних очікувань μ_y і матриці коваріацій $\sum_y$, задані для кожного класу $y \in \{1, \ldots, C\}$.

Оцінка параметрів (за методом максимальної правдоподібності) і їх кількість.

$$\mu_y = \frac{1}{m}\sum_{i=1}^{m} x_i,$$

$$\sum_y = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_y)(x_i - \mu_y)^T,$$

де

m – кількість об’єктів, що відносяться до класу y,

$C \cdot D$ параметрів для оцінки $\mu_y, y \in \{1, \ldots, C\}$,

μ_y – це вектор довжини D.

Всього C класів, відповідно C центрів і $C \cdot D$ параметрів.

$\frac{C \cdot D \cdot (D+1)}{2}$ параметрів для оцінки $\sum_y$.

$\sum_y$ – симетрична матриця, необхідно тільки задати $\frac{D \cdot (D+1)}{2}$ параметрів. Таких матриць всього C по кількості класів. Ще C параметрів знадобиться для того, щоб задати всі апріорні розподілення $p\,(y), y \in \{1, \ldots, C\}$.

В результаті $\frac{C \cdot D \cdot (D+3)}{2} + C$ параметрів містить модель Гаусовського класифікатора без спрощуючих припущень.

Оцінка апостеріорної ймовірності.

Оцінимо логарифм апостеріорної ймовірності:

$$\log p(y|x) = \log p(x|y) + \log p(y) - \log p(x) =$$
$$= -\frac{1}{2}(x-\mu_y)^T\sum_y^{-1}(x-\mu_y) - \frac{1}{2}\log|\sum_y| - \frac{D}{2}\log 2\pi + \log\ p(y)\ - \log p(x).$$

Дискримінантна функція (отримується з останнього виразу після відкидання членів, що не залежать від класу y) має вигляд:

$$g_y(x) = \log p(y) - \frac{1}{2}\log|\sum_y| - \frac{1}{2}(x-\mu_y)^T\sum_y^{-1}(x-\mu_y).$$

2.4 Основні принципи навчання математичної моделі

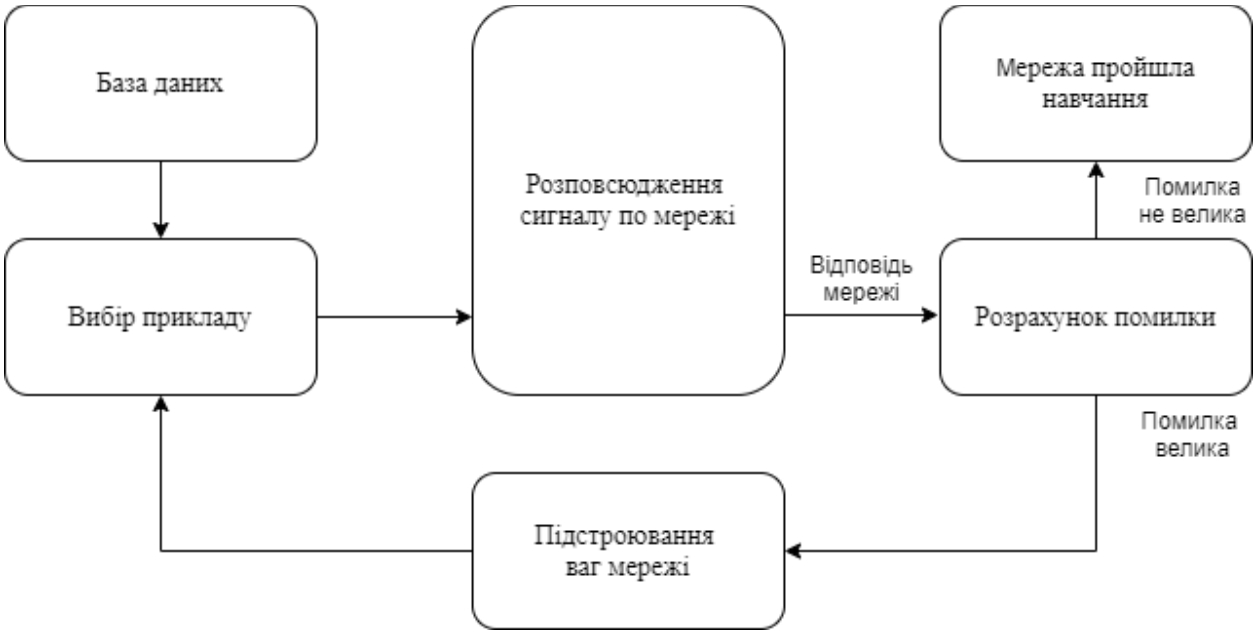


Рисунок 2.9 - Схема навчання мережі

Навчання передбачає, що інформаційна система розпізнає шаблони на прикладах. Ці шаблони містяться в даних. Система вчиться, знаходячи схожі дані серед великих масивів даних. Чим більше даних надається для навчання,

тим швидше йде процес навчання. Важливо враховувати тип даних, за допомогою яких буде проходити навчання, так як точність отриманих даних сильно впливає на успіх. Коли у системи достатньо даних вона здатна робити прогнози.

Це відбувається в 3 етапи:

1) Аналіз даних

2) Знаходження шаблонів

3) Передбачення на основі шаблону

Аналіз даних передбачає збір максимально можливої кількості необхідної точної інформації для навчання системи. Далі необхідно знайти шаблони чи ознаки, за якими система зможе шукати дані в масиві. Далі вибирається метод, від якого буде залежати точність, швидкість роботи і розмір готової моделі.

Розглянемо детальніше алгоритми і методи машинного навчання.

Один з методів машинного навчання - Класифікація: розподілення об'єктів по категоріям за певними ознаками.

На сьогодні використовують для:

— Спам-фільтрів;

— Визначення мови;

— Пошуку схожих документів;

— Аналізу тональності;

— Розпізнавання рукописних букв і цифр;

— Визначення підозрілих транзакцій.

До цього методу можна віднести такі алгоритми, як Наївний Байєсовський алгоритм, Дерева рішень, Логістична регресія, K-найближчих сусідів, Метод опорних векторів.

Для класифікації завжди потрібен учитель - розмічені дані з ознаками і категоріями, які машина буде вчитися визначати за цими ознаками.

Розглянемо приклад класифікації. Припустимо, що хтось хоче взяти кредит в банку. Для того, щоб банк знав, чи здатні ви виплатити його, води

використовують дані про клієнтів. В даних вказаний вік, освіта, місце роботи, розмір заробітної плати і т.д. Проводиться аналіз, з якими з попередніх клієнтів виникали проблеми, а з якими ні, враховуючи ці дані. Для даної задачі використовують дерево рішень. Система розподіляє дані на запитання, на які можна відповісти або «так» або «ні». Тобто, будуть дивитися кредитну історію, гарна вона чи погана, чи є борг чи немає і т.д.

Найпопулярнішим методом Класифікації є Метод опорних векторів. Він шукає, як так провести дві прямі між категоріями, щоб між ними утворився найбільший зазор.

За допомогою Класифікації можливо досить просто знаходити аномалії, тому що коли ознаки об’єкта не відповідають жодній з категорій, це буде одразу помітно.

Ще один метод машинного навчання це Регресія.

Використовується для:

— Прогнозу вартості цінних паперів;

— Аналізу попиту, обсягу продажів;

— Медичні діагнози;

— Будь-які залежності числа від часу.

Регресія дуже схожа на класифікацію, різниця в тому що прогнозується не категорія а число. За допомогою цього методу легко вирішуються будь-які задачі, що залежать від часу (вартість автомобіля по його пробігу, кількість пробок по часу доби, обсяг попиту на товар від зростання компанії і.т.д.).

Принцип в тому, щоб намалювати лінію між точками, яка відобразить залежність. Якщо лінія пряма, то Регресія лінійна, якщо крива, то поліноміальна.

Для того, щоб передбачити, в який час доби на дорогах будуть пробки Регресія ідеально підходить (рисунок 2.10).

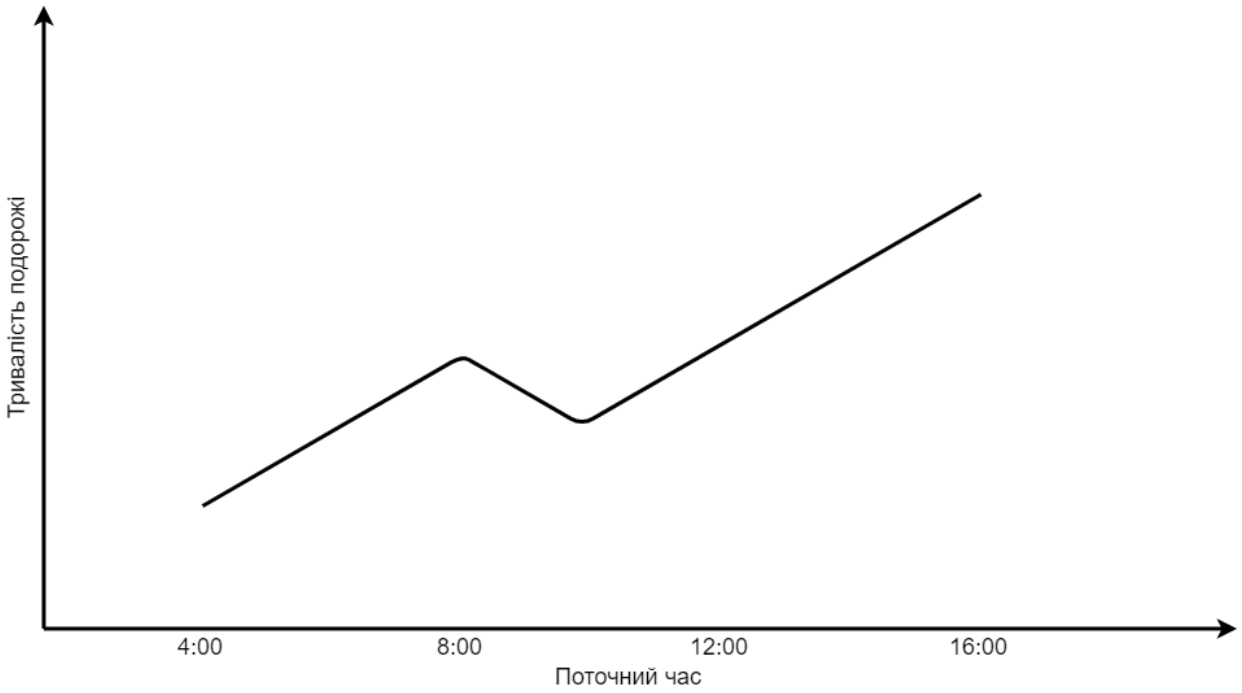


Рисунок 2.10 – Приклад поліноміальної регресії

Наступний метод машинного навчання це Кластеризація – розподіляє об’єкти по невідомим ознакам.

Сьогодні використовують для:

— Сегментації ринку (типів покупців, лояльності);

— Об'єднання близьких точок на карті;

— Стиснення зображень;

— Аналізу і розмітки нових даних;

— Детектори аномального поведінки.

До алгоритмів кластеризації можна віднести алгоритм K-середніх.

Кластеризація це те саме що класифікація, але категорії /кластери заздалегідь не відомі. Кількість кластерів можна задати заздалегідь, або запропонувати це системі. Об’єкти система розподіляє по кластерам використовуючи ознаки, які будуть їй запропоновані.

Стиснення зображень, це одна з задач, які вирішуються методом Кластеризації. При збереженні зображення в форматі PNG можна встановити палітру, наприклад, 32-х кольорову. Тоді система буде знаходити всі

приблизні кольори, наприклад червоний і його відтінки, розрахувавши середній з них, замінить всі на один. Менша кількість кольорів гарантовано дасть менший об’єм файлу. Метод K-середніх часто використовується для вирішення задачі стиснення зображень, для точності визначення кольорів. Принцип такий: випадковим чином розподіляються на палітру кольорів 32 точки, назвемо їх центроїдами. Всі інші точки відносимо до найближчого центроїду від них, отримуючи щось схоже на сузір'я з найближчих кольорів. Потім центроїд треба рухати в центр відповідного сузір'я і повторювати поки центроїди не перестануть рухатися. В результаті кластери виявлені, стабільні і їх рівно 32 як і було необхідно.

Наступний метод - Зменшення Розмірності (Узагальнення) - збирає конкретні ознаки в абстракції вищого рівня.

Сьогодні використовують для:

— Рекомендаційні Системи;

— Красиві візуалізації;

— Визначення тематики та пошуку схожих документів;

— Аналіз фейковий зображень;

— Ризик-менеджмент.

До цього методу можна віднести наступні алгоритми: Метод головних компонент (PCA), Сингулярне розкладання (SVD), Латентний розміщення Дирихле (LDA), Латентно-семантичний аналіз (LSA, pLSA, GLSA), t-SNE (для візуалізації).

Об'єднуючи декілька ознак можна отримати абстракцію.

Розглянемо приклад розподілу документів за темами (Латентно-семантичний аналіз). Для цього завдання будується матриця частоти використання якогось конкретного слова в документах (Таблиця 2.1), в результаті виходить матриця з документами, розсортованими за темами (Таблиця 2.2).

Таблиця 2.1 - Матриця частоти використання слів в документах

	Doc 1	Doc 2	Doc 3
Ноутбук			
Авокадо			
Закон			
Їжа			
Наказ			
Програма			

Таблиця 2.2 – Матриця відсортованих документів за темами

	Doc 1	Doc 2	Doc 3
Ноутбук			
Програма			
Їжа			
Авокадо			
Наказ			
Закон			

Ще один метод навчання - Пошук правил (асоціація). За допомогою цього методу проводиться пошук закономірності в потоці замовлень.

Сьогодні використовують для:

— Прогнозу акцій і розпродажів;

— Аналізу товарів, що купуються разом;

— Розстановки товарів на полицях;

— Аналізу патернів поведінки на веб-сайтах.

Наприклад, покупець в магазині взяв якийсь товар, який знаходиться в дальньому кутку магазину. По дорозі до каси йому можна запропонувати супутній товар.

Навчання з підкріпленням.

— Використовують для:

— Самоврядних автомобілів;

— Роботів пилососів;

— Ігор;

— Автоматичної торгівлі;

— Управління ресурсами підприємств.

Навчання з підкріпленням використовують там, де завданням стоїть не аналіз даних, а виживання в реальному середовищі.

Нейронні мережі і глибоке навчання.

Сьогодні використовують для:

— Може замінити всі алгоритми, описані вище;

— Визначення об'єктів на фото і відео;

— Розпізнавання і синтез мови;

— Обробка зображень, перенесення стилю;

— Машинний переклад.

2.5 Висновки

В другому розділі дипломної роботи були розглянуто що таке машинне навчання, алгоритми машинного навчання математичних моделей, способи використання алгоритмів для навчання мереж Байєса, на основні принципи навчання математичних моделей.

3 МЕРЕЖІ БАЙЄСА ДЛЯ ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Алгоритм опису мінімальної довжини

З розглянутих алгоритмів навчання математичної моделі був обраний алгоритм ОМД (Принцип опису мінімальної довжини).

Принцип ОМД говорить, що серед безлічі моделей Байєсовської мережі необхідно вибрати ту, за допомогою якої можливо описати дані максимально коротко, без втрати інформації.

Задача ОМД виглядає наступним чином:

— в першу чергу задається множина даних для навчання

$D = \{d_1, \dots, d_n\}, d_k = \{x_k^{(1)} x_k^{(2)} \dots x_k^{(N)}\}$, де нижній індекс відповідає за номер спостереження, а верхній індекс відповідає за номер змінної;

— n - кількість спостережень, де кожне спостереження складається з N $(N \geq 2)$ змінних $\mathrm{X}^1, \mathrm{X}^2, \dots, \mathrm{X}^N$;

— кожна $j-$ та змінна, де $j = (1, \dots, N)$, має

$A^{(j)} = \{0,1, \dots, \alpha^{(j)} - 1\} (\alpha^{(j)} \geq 2)$ станів;

— кожна структура $g \in G$ Байєсовської мережі представлена N множинами батьків $\Pi^{(1)}, \dots, \Pi^{(N)}$, тобто для кожного вузла

$j = 1, \dots, N, \Pi^{(j)}$ – це множина батьківських вершин, що $\Pi^{(j)} \subseteq \{\mathrm{X}^1, \dots, \mathrm{X}^N\} \backslash \{X^{(j)}\}$.

Тоді ОМД структури $g \in G$ при заданій послідовності з n спостережень $x^n = d_1, d_2, \dots, d_n$ обчислюється за формулою:

$$L(g, x^n) = H(g, x^n) + (k(g)/2) In(n),$$

де

$k(g)$ це кількість незалежних умовних ймовірностей в мережевій структурі g, а $H(g, x^n)$ – емпірична ентропія:

$$H(g, x^n) = \sum_{j=J} H(j, g, x^n);$$

$$k(g) = \sum_{j=J} k(j, g).$$

ОМД $j-$ї вершини обчислюється за формулою:

$$L(j, g, x^n) = H(j, g, x^n) + (k(j, g)/2)In(n),$$

де

$k(j, g)$ – кількість незалежних умовних ймовірностей $j-$ї вершини:

$$k(j, g) = (\alpha^{(j)} - 1) \prod_{k \in \varphi(j)} \alpha^{(k)}.$$

Тут $\varphi(j) \subseteq \{1, \dots, j-1, j+1, \dots, N\}$ – це така множина, що $\prod^{(j)} = \{X(k): k \in \varphi(j)\}$.

Емпірична ентропія $j-$ї вершини обчислюється за формулою:

$$H(j, g, x^n) \sum_{s \in S(j,g)} \sum_{q \in A^{(j)}} -n[q, s, j, g] In \frac{n[q,s,j,g]}{n[s,j,g]},$$

де

$n[s, j, g] = \sum_{i=1}^{n} I\left(\pi_i^{(j)} = s\right);\ n[q, s, j, g] = \sum_{i=1}^{n} I\left(x_i = q, \pi_i^{(j)} = s\right),$

$\pi^{(j)} = \prod^{(j)}$ означає $X^{(k)} = x^{(k)}, \forall k = \varphi(j)$, функція $I(E) = 1$, коли предикат $E = true$, інакше $I(E) = 0$.

Для навчання БС з використанням методу ОМД необхідно провести перебір всіх можливих нециклічних мережевих структур. В g^* зберігається оптимальна мережева структура. Оптимальною є та, у якої найменше значення функції $L(g, x^n)$. Розглянемо приклад знаходження ОМД для БМ. Нехай маємо деяку довільну вибірку:

Таблиця 3.1 - Вибірка

	1	2	3	4	5	6	7	8	9	10
A	0	0	1	1	1	0	1	1	1	0
B	1	0	0	1	0	0	1	0	1	0
C	1	1	1	1	0	1	0	1	1	1

ОМД будемо обчислювати для БМ, яка зображена на рисунку 3.2.

В даній БМ вершина А не має батьків, тобто $\prod^A = \emptyset$.

Складемо таблицю значень параметрів А.

Таблиця 3.2 – Таблиця значень параметрів А

A	$n[q, s, j, g]$	$n[s, j, g]$
0	4	10
1	6	

Обчислюємо емпіричну ентропію:

$$H(j = 1, g) = -4 \ln\left(\frac{1}{4}\right) - 6 \ln\left(\frac{6}{10}\right) = 6{,}73.$$

Кількість незалежних умовних ймовірностей

$$k(j = 1, g) = 2 - 1 = 1.$$

Отже, довжина опису вершини А дорівнює

$$L(1, g) = 6{,}7 + \frac{1}{2}\ln(10) = 7{,}8.$$

Вершина В також не має батьківської вершини, отже $\prod^{\mathrm{B}} = \emptyset$.

Таблиця 3.3 – Таблиця значень параметрів В

B	$n[q, s, j, g]$	$n[s, j, g]$
0	6	10
1	4	

Обчислюємо емпіричну ентропію:

$$H(j = 2, g) = -6\ln\left(\frac{6}{10}\right) - 4\ln\left(\frac{4}{10}\right) = 6{,}7.$$

Кількість незалежних умовних ймовірностей

$$k(j = 2, g) = 2 - 1 = 1.$$

Отже, довжина опису вершини А дорівнює

$$L(2, g) = 6{,}7 + \frac{1}{2}\ln(10) = 7{,}8.$$

Вершина C має дві батьківські вершини, тому $\prod^{C} = \{A, B\}$.

Таблиця 3.4 – Таблиця значень параметрів C

A	B	C	$n[q,s,j,g]$	$n[s,j,g]$
0	0	0	0	3
0	0	1	3	
0	1	0	0	1
0	1	1	1	
1	0	0	1	3
1	0	1	2	
1	1	0	1	3
1	1	1	2	

Обчислюємо емпіричну ентропію:

$$H(j=3,g) = \left[-0\ln\left(\frac{0}{3}\right) - 3\ln\left(\frac{3}{3}\right)\right] + \left[-0\ln\left(\frac{0}{1}\right) - 1\ln\left(\frac{1}{1}\right)\right] + \left[-1\ln\left(\frac{1}{3}\right) - 2\ln\left(\frac{2}{3}\right)\right] + \left[-1\ln\left(\frac{1}{3}\right) - 2\ln\left(\frac{2}{3}\right)\right] = 3{,}8.$$

Кількість незалежних умовних ймовірностей

$$k(j=3,g) = (2-1)\cdot 2\cdot 2 = 4.$$

Отже, довжина опису вершини C дорівнює

$$L(3,g) = 3{,}8 + \frac{2}{4}\ln(10) = 8{,}4.$$

Тоді довжина опису БМ дорівнює сумі довжин опису кожної її вершини:

$$L(g) = 7{,}8 + 7{,}8 + 8{,}4 = 24.$$

3.2 Робота з вхідними даними та створення статистичної бази даних.

Модель OSI, яка використовується в інтернет мережі, включає в себе 7 рівнів. DDoS-атаки багаторівневі, тобто можливі на кожному рівні моделі OSI.

Наслідки DDoS-атаки на кожному з рівнів:

— 1-й рівень OSI: фізичний;

- мережеве обладнання стає непридатним і потребує ремонту для відновлення роботи;

— 2-й рівень OSI: канальний;

- потоки даних від відправника одержувачу блокують роботу всіх портів;

— 3-й рівень OSI: мережевий;

- зниження пропускної здатності мережі, що атакується;
- можлива перевантаженість брандмауера;

— 4-й рівень OSI: транспортний;

- досягнення меж по ширині каналу або за кількістю допустимих підключень;
- порушення роботи мережевого обладнання;

— 5-й рівень OSI: сеансовий;

- робить неможливим для адміністратора управління мережевим комутатором;

— 6-й рівень OSI: представницький;

- атаковані системи можуть перестати приймати SSL з'єднання або автоматично перевантажуватися;

— 7-й рівень OSI: прикладний;

- нестача ресурсів;
- надмірне споживання системних ресурсів службами на сервері, що знаходиться під атакою.

Таблиця 3.5 - Опис аномалій мережевого трафіку

Тип аномалії і причина виникнення	Опис аномалії	Зміни у мережевому трафіку
Альфа-аномалія	Дуже високий рівень трафіку типу точка-точка.	Викид в поданні трафіку байти / с, пакети / с по одному домінуючому потоку джерело-призначення. Тривалість, як правило, не перевищує 10 хвилин.
DDoS, DoS атаки	Атака типу «відмова в обслуговуванні» на одну ціль.	Викид в поданні трафіку пакети / с, потоки / с, від безлічі джерел до однієї цілі.
Перевантаження	Часті звернення до одного мережевого ресурсу або сервісу.	Стрибок трафіку по потокам / с до однієї домінуючої IP-адреси домінуючому порту. Короткочасна аномалія.
Сканування мережі / портів	Сканування мережі за певним відкритим портом або сканування одного хоста по всіх портах з метою пошуку	Стрибок трафіку по потокам / с, з декількома пакетами в потоках від одної домінуючої IP-адреси.

	вразливостей в ІС.	
Діяльність черв’яка	Зловмисне ПЗ, яке здатне самостійно поширюватися по мережі і використовує уразливості ІС.	Викид в трафіку без домінуючої адреси призначення, завжди з одним або декількома домінуючими портами призначення.
Точка-мультиточка	Поширення контенту від одного сервера до багатьом користувачам.	Викид в пакетах, байтах від домінуючого джерела до кількох цілей, до одного порту.
Відключення	Падіння в трафіку між парою джерело-призначення.	Падіння трафіку по пакетам, потоках, байтах до нуля. Тривалість може бути довгою. Може включати всі пари джерело-призначення від або до одного маршрутизатора.
Перемикання потоку	Перемикання потоків трафіку з одного вхідного маршрутизатора на інший вхідний маршрутизатор.	Падіння в байтах або пакетах в одному потоці трафіку і викид в іншому. Може поширюватися на кілька потоків.

Таблиця 3.6 - Атаки по рівням моделі OSI

Багаторівнева атака	DDoS, DoS атаки
Рівень додатків	Відмови, спотворення даних
Транспортний рівень	Імітація сесії, SYN-flood
Мережевий рівень	Черв'яки, Трояни, споживання ресурсів системи
Канальний рівень	Аналіз трафіка, уразливість WEP шифрування
Фізичний рівень	Перешкоди (DoS в бездротових мережах), перехват,

Представлені вище дані можна використати в якості параметрів для побудови БС дослідницької роботи (рисунок 3.1).

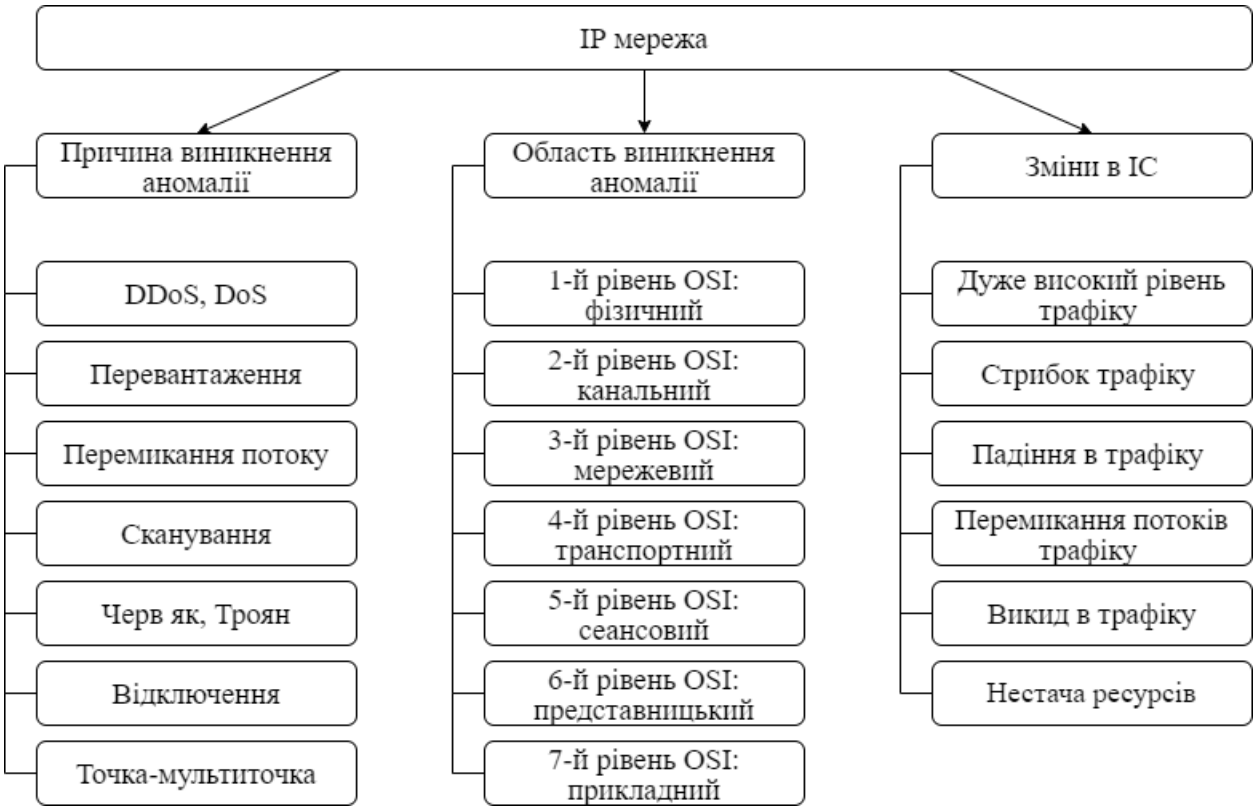


Рисунок 3.1 – Графічне зображення параметрів БМ відповідно до вузлів

3.3 Побудова математичної моделі на базі мереж Байєса.

Послідовність побудови Байєсовської мережі:

1. Провести аналіз процесу, зібрати данні;

2. Створення статистичної бази даних;

3. Згенерувати вузли та дуги Байєсовської мережі;

4. Провести розрахунок апріорних ймовірностей для вершин графа та оптимізувати побудовану мережу;

5. Провести навчання мережі за допомогою обраного алгоритму;

6. Використати мережу для поставленої задачі і проведення розрахунків;

7. Вивести результати розрахунків користувачеві.

Визначення параметрів це необхідність виставити апріорні розподіли для вузлів, у яких немає батьківських вершин, а також розподілу умовних ймовірностей для всіх інших вузлів Байєсовської мережі. Параметри можуть бути задані експертом, чи можуть бути отримані з даних. Можна комбінувати ці підходи. Апріорний розподіл ймовірностей - це безумовний розподіл, який не змінюється, при цьому не важливо яким чином були отримані свідчення. Кожна змінна може приймати необмежену кількість значень, але велике число значень ускладнює модель. Тому в Байєсовських мережах часто використовують бінарні змінні.

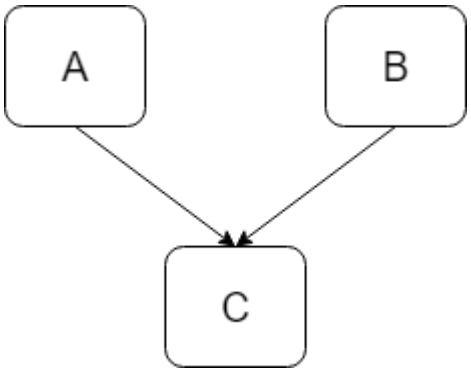


Рисунок 3.2 - Найпростіша Байєсовська мережа

Розглянемо принцип побудови Байєсовської мережі на прикладі. У прикладі представлена найпростіша Байєсовська мережа, яка складається лише з трьох вузлів. Припустимо, що в змодельованій Байєсовській мережі вершини можуть приймати лише два стани і перебувати в одному з них. У таблиці 3.7 представлені можливі стани для вершин побудованого графа.

Таблиця 3.7 – Таблиця можливих станів вершин графа

Вузол	Стан 1	Стан 2
A	1	0
B	1	0
C	1	0

Вершина A знаходиться в стані 1, якщо подія сталася, інакше вона знаходиться в стані 0. Це аналогічно і для інших вершин. Дана Байєсовська мережа показує, що є причинно-наслідкова залежність від події A до події C і від події B до події C, що відображається стрілками. При причинно-наслідковій залежності між вершинами, наприклад від вершини A до вершини C, стан, в якому перебуває вершина A впливає на стан вершини C. На рисунку 2.4 зображено графічне представлення даної Байєсовської мережі. Однак графічне представлення графа не можна назвати Байєсовською мережею, а тому необхідно розрахувати її якісне уявлення. Для цього складемо таблиці умовних ймовірностей для вершин графа. Розраховані ймовірності відображені в таблицях 3.8, 3.9, 3.10. Представлені таблиці демонструють ймовірність перебування вершини графа в певному стані, який, в свою чергу, залежить від стану батьківської вершини, якщо така існує. У разі, якщо вузол графа не має батьківських вершин, в такому випадку ймовірності цього вузла є маргінальними, тобто не залежать ні від чого.

Таблиця 3.8 – Таблиця ймовірностей для вершини A

Апріорна ймовірність p (A)	
A=1	A=0
0.1	0.9

Таблиця 3.9 – Таблиця ймовірностей для вершини B

Апріорна ймовірність p (B)	
B=1	B=0
0.1	0.9

Таблиця 3.10 – Таблиця ймовірностей для вершин C, A, B

Таблиця умовних ймовірностей p (C | A, B)				
	B=1		B=0	
	A=1	A=0	A=1	A=0
C=1	0.95	0.85	0.90	0.02
C=0	0.5	0.15	0.10	0.98

Для опису Байєсовської мережі необхідно визначити параметри вузлів графа, а так само їх зв'язки, що інакше можна назвати топологією графа. Параметри вузлів можливо отримати з даних, які використовуються для навчання, але розрахувати правильну топологію складніше. У разі, якщо деякі вузли нам невідомі або приховані, або є неповні дані, потрібні особливі підходи. Існують чотири випадки навчання мережі, які представлені в таблиці 3.11.

Таблиця 3.11 – Випадки навчання Байєсовської мережі

Структура	Спостереження	Метод
Відома	Повне	Максимально можлива оцінка правдоподібності
Відома	Часткове	Максимізація математичного очікування або жадібний метод пошуку екстремуму
Невідома	Повне	Пошук в просторі моделей
Невідома	Часткове	Структурний алгоритм максимізації математичного очікування або стиснення кордонів

У разі, якщо відома структура і повна спостережуваність, тоді обчислюються параметри для кожного умовного імовірнісного розподілу, які максимізують правдоподібність навчальних даних. Це найпростіший випадок.

У разі, коли відома структура, але є часткова спостережуваність, тобто коли деякі з вузлів приховані, можна застосувати алгоритм максимізації математичного очікування (ММО), для знаходження локальної оптимальної оцінки максимальної правдоподібності (ОМП) параметрів.

Алгоритм ММО передбачає, що якщо нам відомі значення всіх вузлів, навчання на кроці М є простим, так як ми знайомі з попередніми. Тоді на кроці Е, ми обчислюємо очікувані значення вузлів, що використовують алгоритм виведення, і потім використовуємо ці значення так, як ніби вони були отримані зі спостережень.

Якщо структура невідома і спостережуваність повна, тоді найбільш вірогідною моделлю в даному випадку є повний граф, тому що в цьому випадку буде задіяно найбільшу кількість параметрів. Після вибору

структури наступним кроком є навчання структури. Це завдання є завданням з нелінійною поліноміальною оцінкою кількості ітерацій. Тому зазвичай використовують локальні алгоритми пошуку, такі як жадібний алгоритм методу пошуку екстремуму або метод гілок і меж.

Найскладнішим випадком вважається, якщо невідома структура і спостережуваність часткова. Присутні приховані змінні і некоректні дані. В такому випадку доцільно використовувати структурний алгоритм максимізації математичного очікування (СММО) або алгоритм стиснення кордонів.

Алгоритм СММО з'єднує в собі алгоритм ММО, який проводить оптимізацію параметрів, зі структурним пошуком моделі відбору. Цей алгоритм навчає мережі, ґрунтуючись на штрафних імовірнісних значеннях, які включають значення, отримані за допомогою Байєсовського інформаційного критерію, принципу мінімальної довжини опису, а також значення інших критеріїв.

Метод стиснення кордонів моделює відсутність даних, припускаючи що ймовірність відсутності даних знаходиться в інтервалі від 0 до 1. Тобто проводитися обчислення цього інтервалу, за наявною інформацією. Після цього проводиться стиснення кордонів інтервалу в точку за допомогою використання опуклою комбінації точок екстремумів, ґрунтуючись на інформації про неповні дані.

Імовірнісний вивід це одне з найпоширеніших завдань, яке вирішується з використанням Байєсовської мережі.

Для побудови та візуалізації Байєсовської мережі існують програмні засоби, в яких є необхідні інструменти побудови і можливість введення нових даних в мережу і отримання нових рішень за допомогою перерахунку ймовірностей.

3.4 Висновки

В третьому розділі дипломної роботи було розглянуто і детально описано один з алгоритмів навчання математичної моделі. Були розглянуті відомі мережеві аномалії, причини та області їх виникнення в мережі, а також їх вплив на роботу інформаційних систем. Був складений перелік параметрів інформаційної системи для спостереження в системі виявлення зловмисного програмного забезпечення з метою виявлення вторгнень. Також був розглянутий приклад побудови найпростішої Байєсовської мережі та особливості застосування алгоритму ОМД для навчання математичної моделі.

4 ВІЗУАЛІЗАЦІЯ РОБОТИ МАТЕМАТИЧНОЇ МОДЕЛІ

4.1 Побудова Байєсовської мережі для використання в системах виявлення зловмисного програмного забезпечення

Для використання Байєсовської мережі в системі виявлення зловмисного програмного забезпечення необхідно визначити вузли системи, які будуть перевірятися і спостерігатися СВЗПЗ на наявність аномалій, скласти базу даних аномалій, які можуть виникнути в ІС, а також визначити причини (фактори) виникнення аномалій у вузлах системи. Для появи, наприклад, двох різних аномалій в ІС можуть служити загальні фактори.

У представленій мережі наслідком впливу деяких факторів на ІС є аномалії в вузлах системи, що означає зараження ЗПЗ вузлів системи, в наслідок чого зараженою є система в цілому. Це графічно зображено на малюнку 4.1. В наслідок проведення розрахунків за допомогою складеної моделі передбачається виведення ймовірності відповідності фактору (вірусу) сигнатурі з бази сигнатур вірусів або сигнатурі з бази аномалій.

В наслідок отриманої інформації про системи виявлення аномалій, можна зробити висновок, що необхідно робити «зліпок» нормальної поведінки вузлів системи, які будуть спостерігатися СВЗПЗ, далі розділяти аномалії на дві категорії: на сигнатури атаки, вже відомі і використовуються в системах виявлення зловживань і на аномалії, які не відповідають ні однієї з відомих. У другому випадку ці атаки розділяти так само на дві категорії: аномалії, які визнаються атакою і атаки, які не вважатимуться аномальними. У разі, якщо є відхилення від нормальної поведінки і сигнатура атаки відома, атаку розглядати як аномальну і робити відповідні дії. Якщо присутній відхилення від нормальної поведінки, але сигнатури атаки в базі не виявлено - розглядати атаку як аномальну. При нормальній поведінці або незначному

відхиленні атаку пропускати. У таблиці 3.1 наочно представлені шляхи вирішення проблеми (1-умова виконано, 0 - умова не виконана).

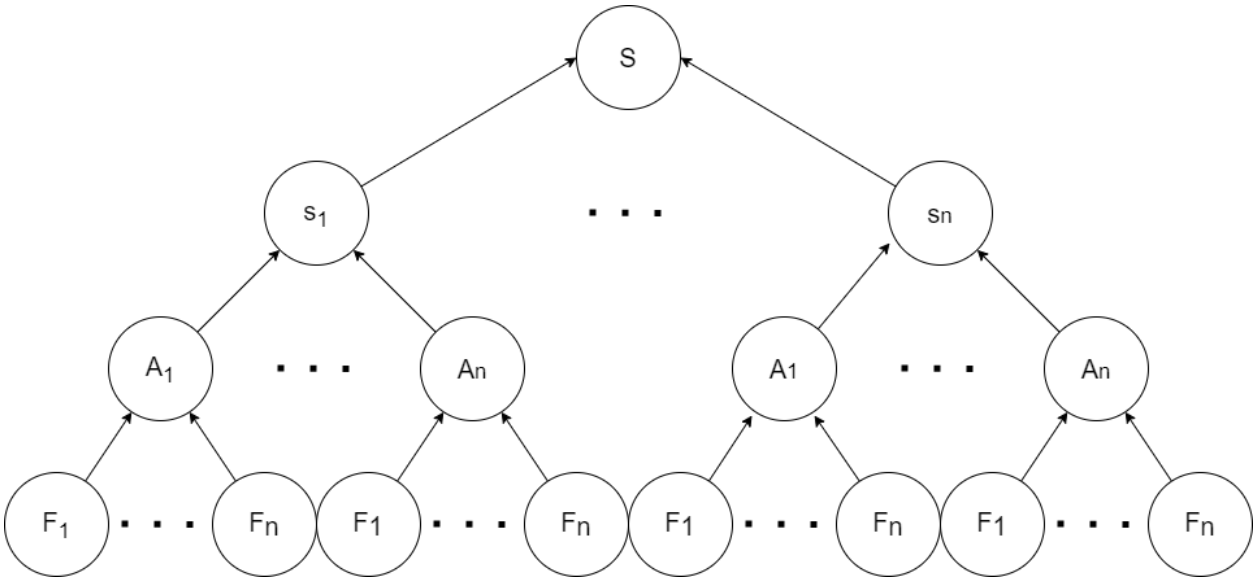


Рисунок 4.1 – Загальний вигляд БМ для використання в СВЗПЗ

де

S – ІС, на якій встановлена СВЗПЗ;

[S_1 – S_n] – множена вузлів ІС, що спостерігаються;

[A_1 - A_n] – множина аномалій;

[F_1 - F_n] – множина факторів.

Таблиця 4.1 – Поведінка моделі при виявленні атаки

Сигнатура атаки	Аномальна атака	Пропуск атаки	Нормальна поведінка
Відома	1	0	0
Невідома	0	1	1
	1	0	0

Розглянемо простий приклад Байєсовської мережі, де в інформаційній системі спостерігається один вузол на наявність аномалії. Якщо виявлена атака визнана аномальною і її сигнатура є в базі, СВЗПЗ видає повідомлення про конкретну атаку і видає варіанти вирішення проблеми. У тому випадку, якщо зафіксована атака невідома, і відхилень немає - атака просто пропускається, при наявності відхилень необхідно провести вивчення аномалії, провести порівняння з наявними в базі сигнатурами (зафіксувати частоту появи аномалії, джерела аномалії і т.д.), простежити вплив на ІС, видати сповіщення про атаку.

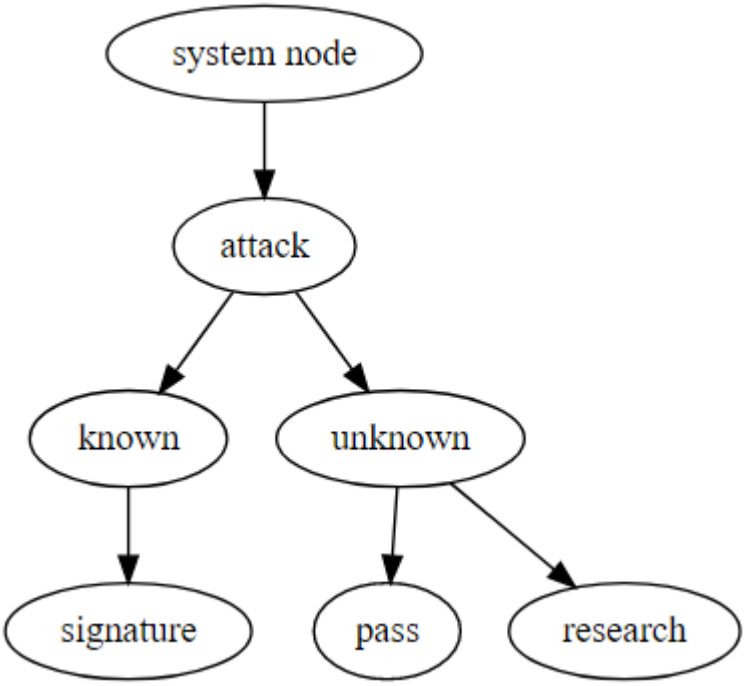


Рисунок 4.2 - Дії СВЗПЗ при виявленні атаки

4.2 Візуалізація Байєсовської мережі в програмі Hugin Lite

Розглянемо приклад побудови БМ для одного з вузлів ІС. В якості вузла, що перевірятиметься буде використана мережа IP. Розглянемо індикатори / ознаки, які мають велике значення для виявлення мережевих атак:

— Джерело виникнення;

— Область виникнення;

— Характер зміни трафіку.

На рисунку 4.3 зображена БМ для даного випадку.

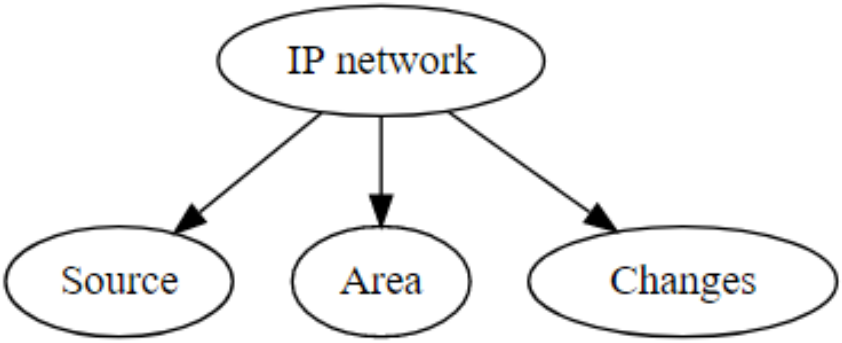


Рисунок 4.3 – БМ для виявлення аномалії в мережі IP

Вершині «IP network» відповідає апріорна ймовірність виникнення аномальної атаки без урахування P (IP network), вершині «Source» відповідає значення ймовірності P (Source|IP network), вершині «Area» - P (Area|IP network) и вершині Changes - P (Changes|IP network).

Розглянемо програмне забезпечення для роботи з Байєсовськими мережами, яке підтримується в даний час.

Так як в Байєсовських мережах візуалізується взаємозв'язок між елементами, які входять в модель, це призводить до ряду вимог. Для початку необхідні засоби для візуалізації результатів розрахунку, що реалізовано у багатьох програмах. Так само будуть корисні такі функції, які дозволять

створювати і досліджувати модель поступово. Дане завдання вирішується складанням структури моделі самостійно з блоків певного виду і покажчиків зв'язку між блоками.

Нижче представлений список найбільш відомих і підтримуваних на даний момент програм для роботи з Байєсовськими мережами з графічним інтерфейсом і які відповідають вказаним вимогам:

— AgenaRisk: на офіційному сайті AgenaRisk сказано, що вони розробляють та продають новаторські продукти, використовуючи Байєсовські мережі. Їх продукт AgenaRisk Desktop - це модель проектування та виконання середовища для мереж Байєса для операційних систем Windows, Linux та Macintosh. За допомогою AgenaRisk Desktop можна створювати індивідуальні моделі для будь-якої проблеми, використовуючи статистичні дані. Моделі, що розроблені в AgenaRisk Desktop, можуть бути інтегровані в більш широкий сервіс за допомогою AgenaRisk Developer і, зрештою, розгорнуті за допомогою AgenaRisk Enterprise. Їх технологія AgenaRisk заснована на 30-річному дослідженні інформатики, штучного інтелекту, Байєсовської ймовірності, статистики та розумних даних. Ця технологія та супутня методологія були опубліковані у провідних академічних журналах зі штучного інтелекту, машинного навчання, актуарії, науки про прийняття рішень та когнітивних наук. На рисунку 3.4 представлений скріншот програми AgenaRisk. Даний продукт платний, а у безкоштовної версії є істотні обмеження, такі як відсутність повної технічної підтримки, обмеження функціональності і неможливість запуску на 64-бітному процесорі;

— BayesiaLab: це програмне забезпечення для Байєсовських мереж, для досліджень, аналітики і міркувань. ПО доступне для операційних систем Windows, macOS, Linux / Unix, забезпечує вченим всебічне «лабораторне» середовище для машинного

навчання, моделювання знань, діагностики, аналізу, моделювання та оптимізації. Цей продукт є результатом більше ніж двадцятирічних досліджень та розробки програмного забезпечення доктора Ліонеля Жуффа та доктора Пола Мунтяну. У 2001 р. Їхні дослідницькі зусилля призвели до утворення Bayesia S.A.S. зі штаб-квартирою в Лавалі на північному заході Франції. Сьогодні компанія є провідним світовим постачальником мережевого програмного забезпечення Байєса, що обслуговує сотні найбільших корпорацій та дослідницьких організацій у всьому світі. На рисунку 3.5 представлений скріншот програми BayesiaLab. Має надзвичайно складний графічний інтерфейс для новачків, а також має дуже високу вартість. Доступна безкоштовна версія на тридцять днів без можливості зберігання побудованих мереж;

— Bayes Server: Це програмне забезпечення для штучного інтелекту для обґрунтування, виявлення, діагностики та автоматизованого прийняття рішень. Дозволяє створювати дані та / або експертні рішення складних проблем за допомогою байєсівських мереж. Технології Bayes Server використовуються в аерокосмічній, оборонній, автомобільній, космічній, машинобудівній, нафтогазовій галузях, охороні здоров'я, фінансах та інших передових галузях. А також використовується для таких задач як: інтелектуальне обслуговування, виявлення аномалій, інтелектуальне виробництво, моделювання ризиків, інтелектуальна програма діагностики / усунення несправностей. Bayes Server дозволяє працювати як з призначеним для користувача інтерфейсом, так і з кросплатформним API. Перша версія цього програмного забезпечення вийшла у 2008 році. На рисунку 3.6 зображений інтерфейс Bayes Server . Bayes Server є платним продуктом. Є дві безкоштовні версії продукту: Bayes Server Express edition, де обмежена кількість вузлів, випадків, потоків і Bayes Server Trial

edition, де сесія обмежена за часом в 120 хвилин і немає можливості зберігати побудовані мережі. В даній програмі можливо працювати тільки з моделями, які побудовані в тільки в ній, не можна імпортувати моделі з інших програмних продуктів.

— GeNIe Modeler: GeNIe Modeler - це графічний користувальницький інтерфейс (GUI) для SMILE Engine. SMILE («Structural Modeling, Inference, and Learning Engine) є незалежною бібліотекою класів C ++, що реалізує моделі, такі як Байєсовські мережі, діаграми впливу і моделі структурних рівнянь. Окремі класи, визначені в SMILE API, дозволяють створювати, редагувати, зберігати і завантажувати графічні моделі і використовувати їх для імовірнісного виведення і прийняття рішень в умовах невизначеності. GeNIe Modeler дозволяє створювати інтерактивні моделі та навчатись. Він написаний для середовища Windows, але може також використовуватися на macOS та Linux під Wine. Використовується з 1998 року та отримав широке визнання як в наукових колах, так і в промисловості, і має тисячі користувачів у всьому світі. З 2015 року ліцензія на цей продукт належить компанії BayesFusion. Забезпечує повну свободу моделювання. Програмне забезпечення має набір приблизних стохастичних алгоритмів вибірки, здатних вирішувати будь-які моделі, створені користувачами. На рисунку 3.7 зображений інтерфейс GeNIe Modeler. Повна інтеграція з MS Excel, вирізання та вставка даних у внутрішній перегляд електронних таблиць GeNIe. Гнучка обробка даних, включаючи імпорт із зовнішніх баз даних. Потужна діагностична функціональність, включаючи значення обчислення інформації, яка впорядковує можливі діагностичні тести та запитання. Підтримує всі основні типи файлів мережі Байєса (наприклад, Hugin, Netica, Ergo). BayesBox і BayesMobile дозволяють використовувати моделі GeNIe через будь-який веб-браузер або мобільний пристрій.

— Hugin Expert: Hugin Expert це програмне забезпечення, що призначене для використання в комерційних та академічних цілях. Має декілька версій: HUGIN Explorer / HUGIN Educational (Використовується для побудови, експериментування та зміни моделей Байєсовської мережі та діаграм впливу. Цей пакет не призначений для розробки програмного забезпечення.), HUGIN Developer / HUGIN Researcher (Цей пакет призначений для створення і вдосконалення додатків або послуг на основі технології HUGIN. Він має гнучкий та зручний графічний інтерфейс користувача (HGUI) та вдосконалений механізм прийняття рішень HUGIN (HDE) для розробки додатків. HDE містить функціонал для використання баз знань у середовищі програмування та постачається з інтерфейсами прикладних програм (API)). HUGIN Explorer та HUGIN Developer мають комерційну ліцензію. Доступна безкоштовна версія продукту HUGIN Lite – це обмежена версія розробника / дослідника HUGIN, може бути використана лише для демонстрації можливостей та підтвердження концепції. HUGIN Lite включає легкий у засвоєнні графічний користувальницький інтерфейс, механізм прийняття рішень HUGIN та чотири API, а також повну бібліотеку заздалегідь створених баз знань з різних галузей бізнесу. Ця версія має обмеження, може обробляти до 50 станів і вивчати до 500 випадків.

Для візуалізації роботи математичної моделі було використане ПЗ Hugin Lite 8.9, тому що цей програмний продукт має зручний користувацький інтерфейс і призначений для демонстрації роботи моделі, що відповідає поставленим задачам.

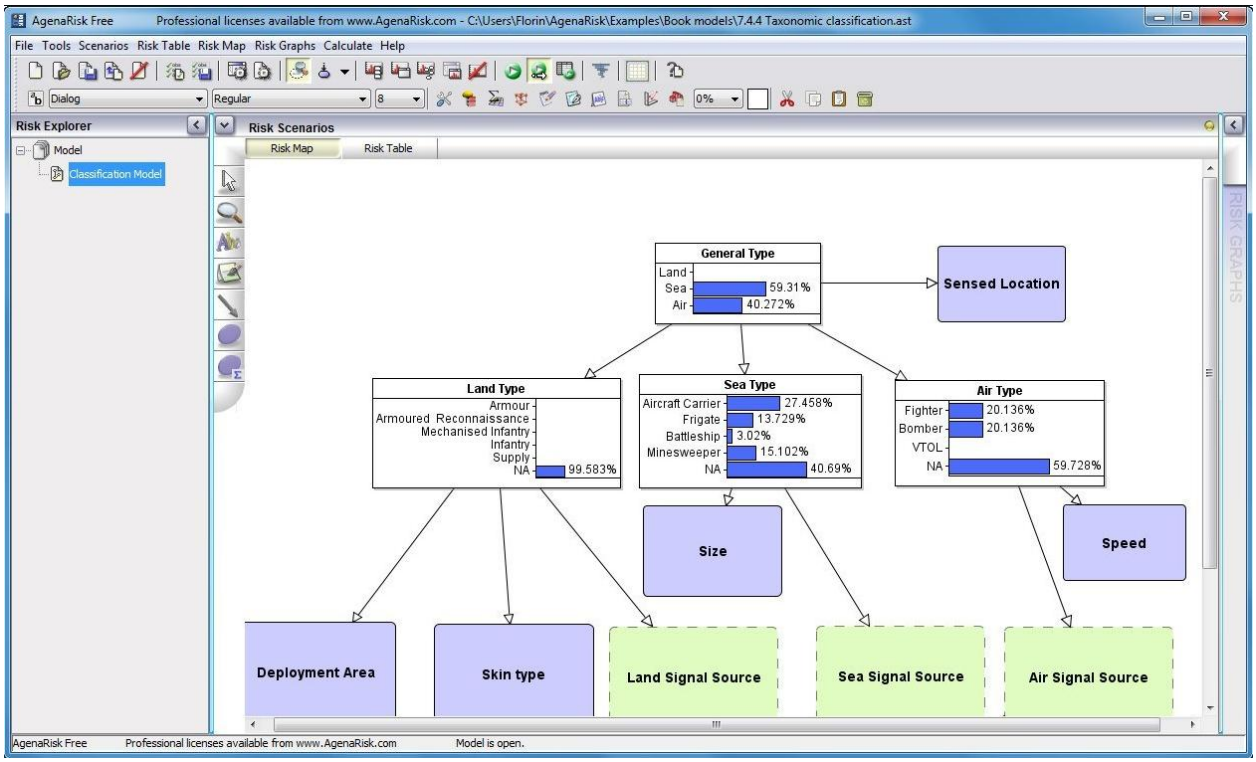


Рисунок 4.4 – Інтерфейс AgenaRisk

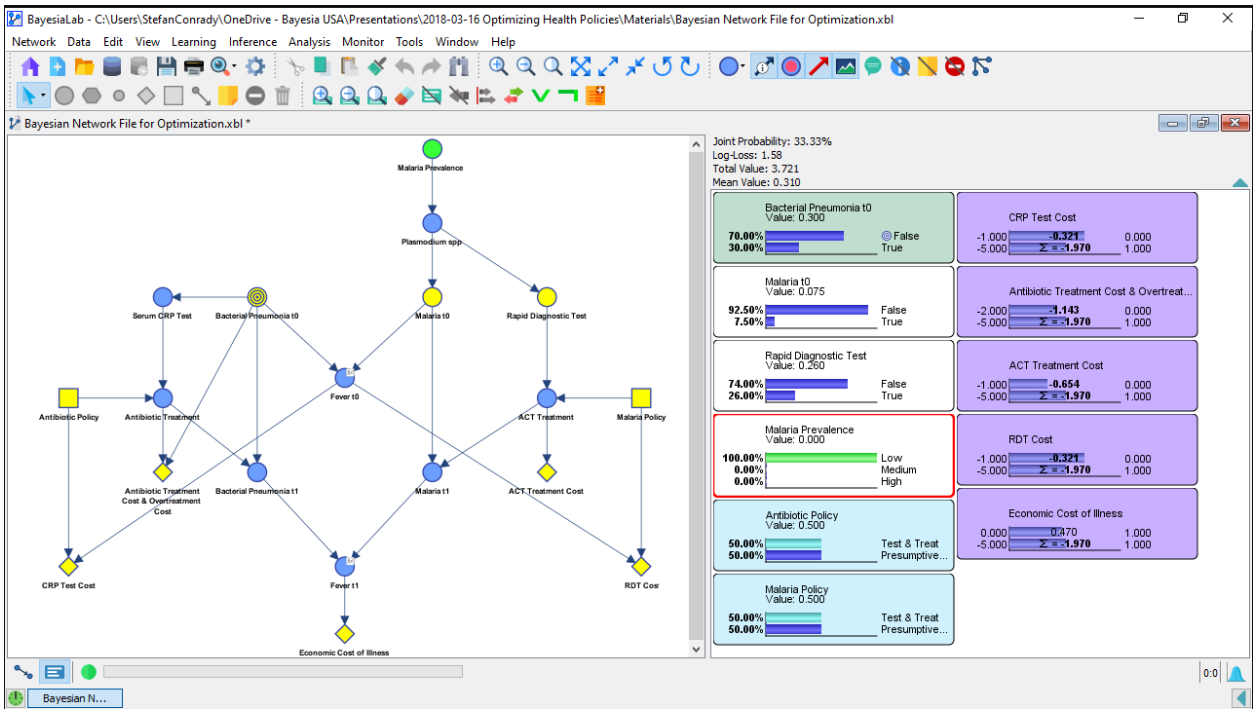


Рисунок 4.5 – Інтерфейс BayesiaLab

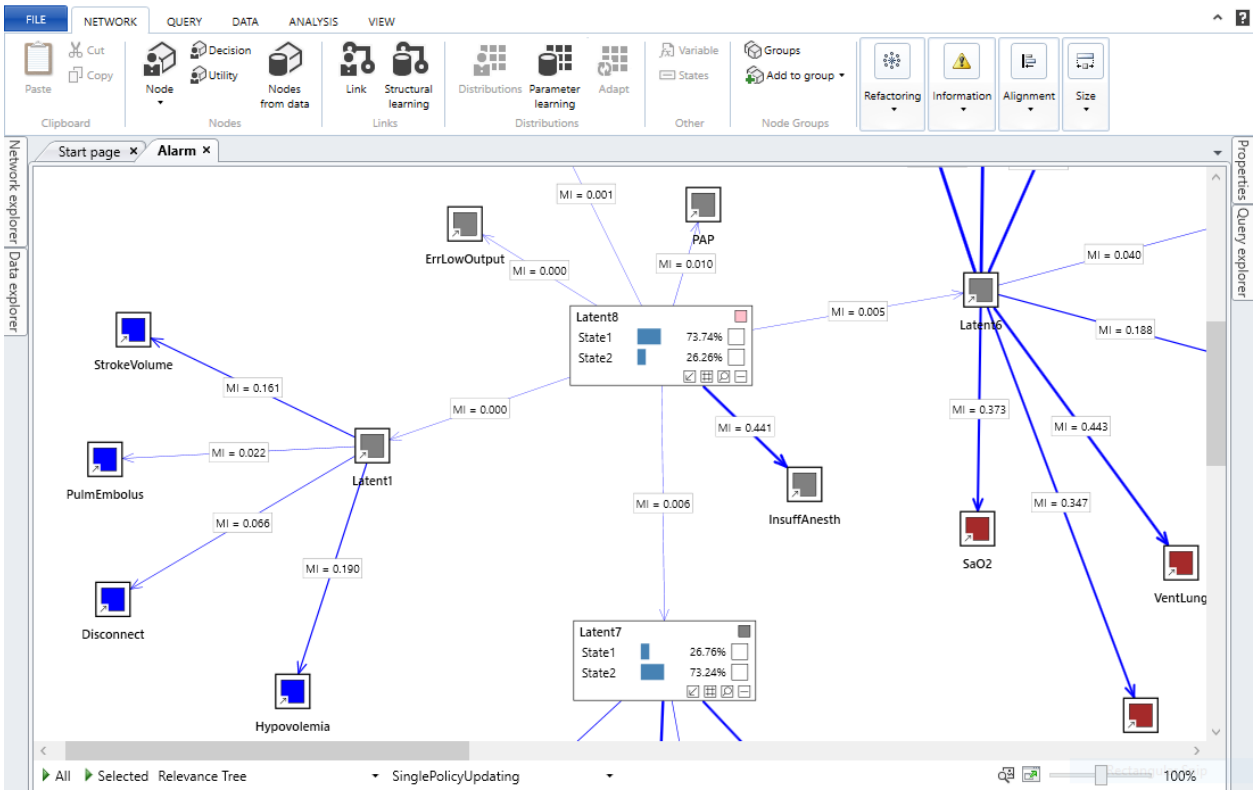


Рисунок 4.6 – Інтерфейс Bayes Server

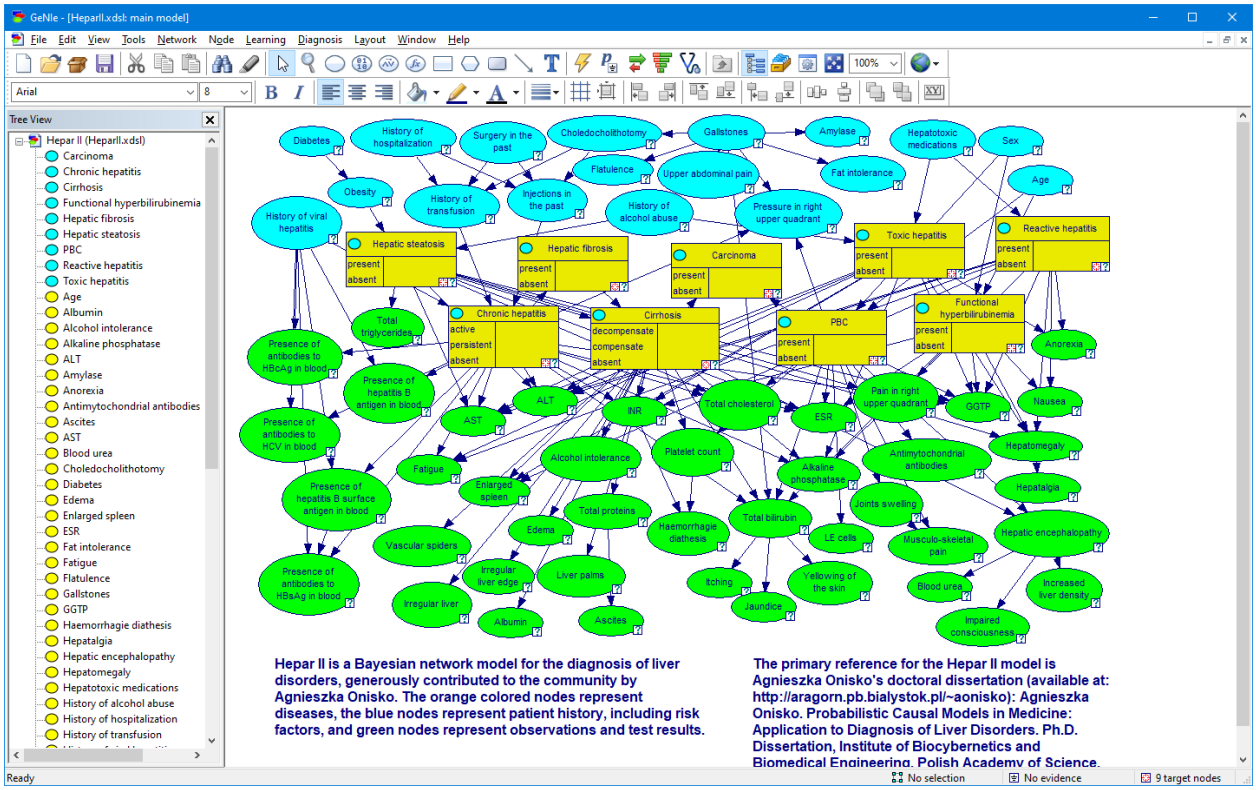


Рисунок 4.7 – Інтерфейс GeNIe Modeler

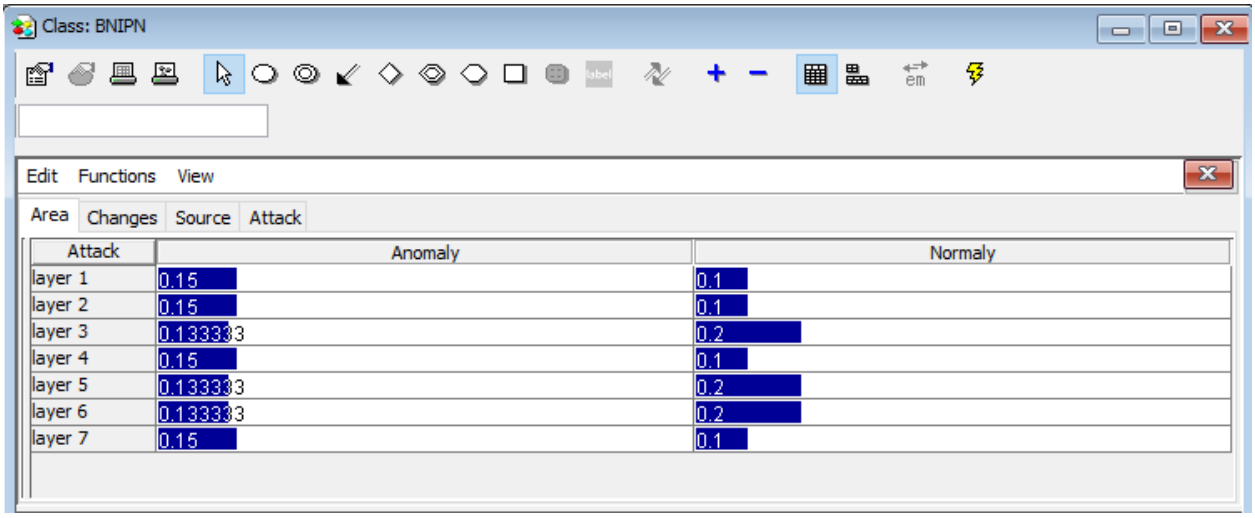


Рисунок 4.8 – Умовні ймовірності P (Area| Attack)

На рисунку 4.8 зображений скріншот таблиці ймовірностей для вузла Area. Так як вершина Area має батьківську вершину Attack, тому її стани залежать від станів Anomaly and Normaly. Ця вершина відповідає за виявлення області виникнення аномалії. Для пунктів перевірки були використані рівні моделі OSI.

На рисунку 4.9 зображений скріншот таблиці ймовірностей для вузла Changes. Даний вузол відповідає за відслідковування змін в інформаційні системі. Для перевірки використані така поведінка:

— Падіння трафіку;

— Розповсюдження контенту;

— Сканування мережі;

— Часті звернення до одного мережевого ресурсу або сервісу;

— Відмова в обслуговуванні;

— Дуже високий рівень трафіку типу точка-точка;

— Нестача системних ресурсів;

— Неможливе управління мережевим комутатором.

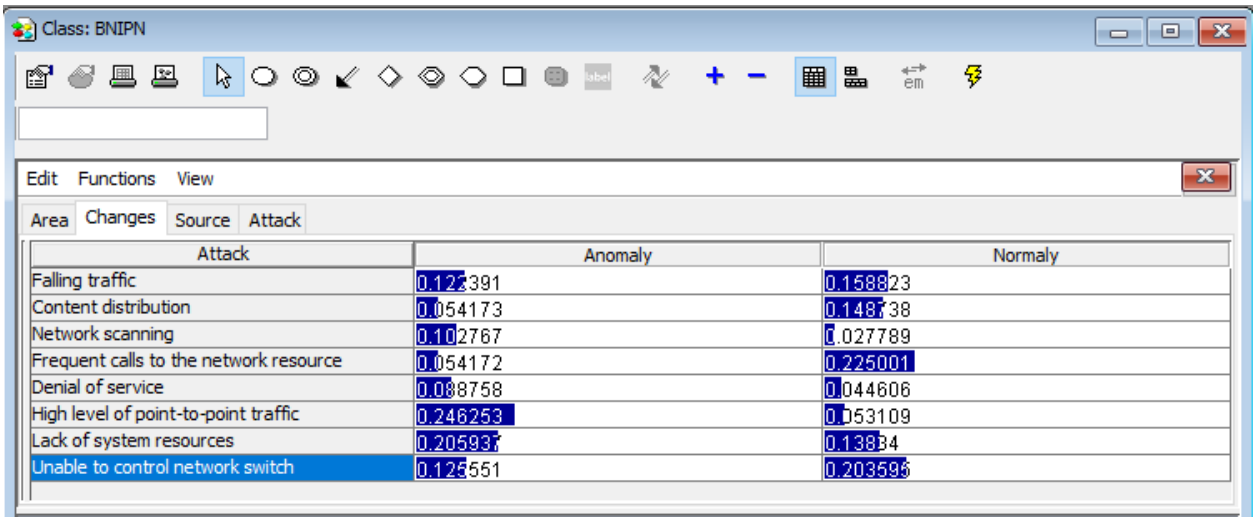


Рисунок 4.9 – Умовні ймовірності P (Changes | Attack)

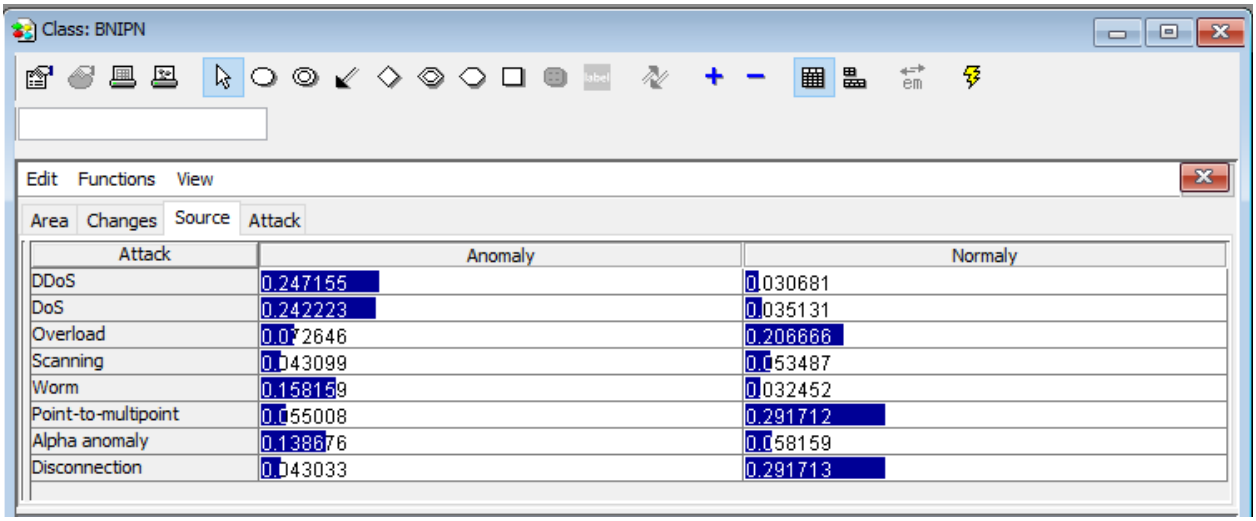


Рисунок 4.10 – Умовні ймовірності P (Source | Attack)

На рисунку 4.10 зображений скріншот таблиці ймовірностей для вузла Source. Даний вузол перевіряє причину виникнення аномалії, для цього були використані такі фактори як:

— DDos і DoS атаки;

— Перевантаження;

— Сканування;

— Черв’як;

— Аномалія типу точка-мультиточка;

— Альфа-аномалія;

— Відключення системи.

На рисунку 4.11 зображений скріншот таблиці ймовірностей для вузла Attack. Вершина Attack є батьківською вершиною для вузлів Source, Changes та Area. Ймовірності цієї вершини є маргінальними. Вершина має два стани: Anomaly – відповідає аномальній поведінці системи / вузла і Normaly, що відповідає встановленій нормальній поведінці ІС.

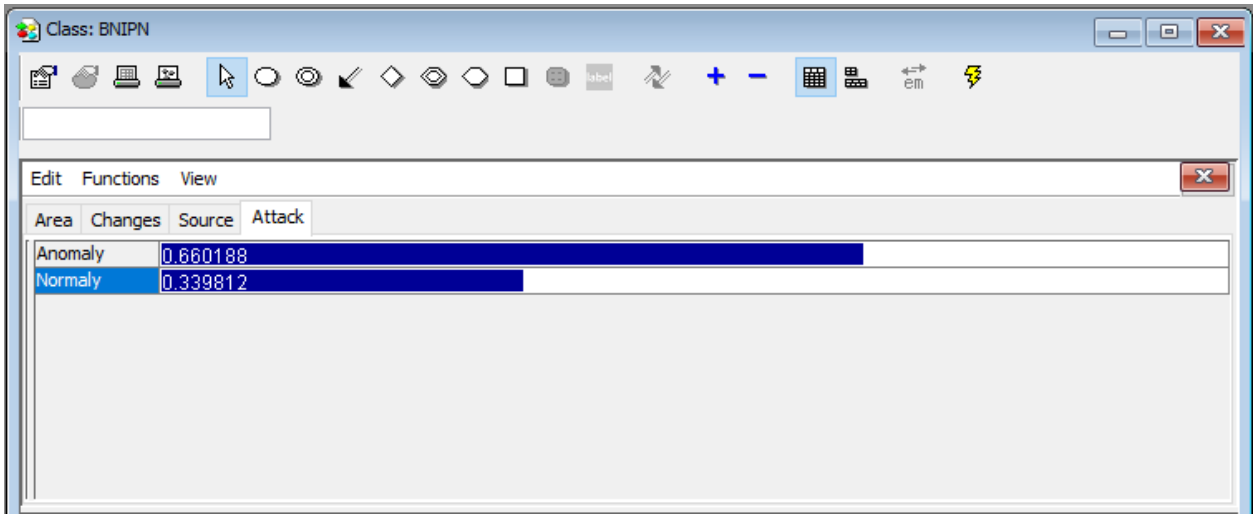


Рисунок 4.11 – Апріорна ймовірність P (Attack)

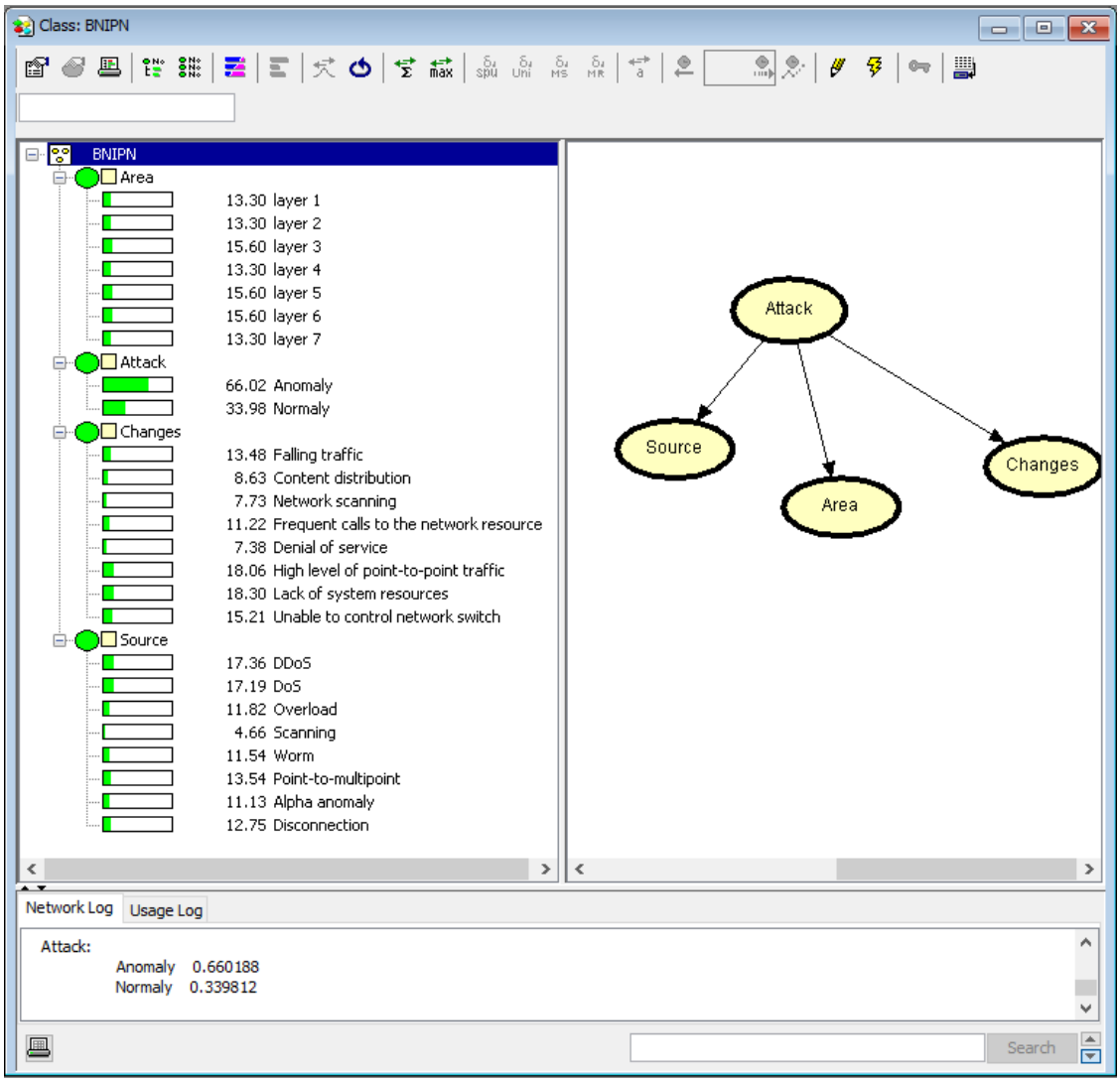


Рисунок 4.12 – Виконання розрахунків

На рисунку 4.12 зображений скріншот програми з результатами обчислення роботи Байєсовської мережі. Зліва на скріншоті продемонстровані отримані ймовірності відповідності атаки до конкретної сигнатури аномалії, ймовірності причини виникнення аномалії та ймовірності області виникнення аномалії.

4.3 Висновки

У четвертому розділі дипломної роботи було сформовано загальній вигляд Байєсовської мережі для використання в системі виявлення зловмисного програмного забезпечення. Так як сформована Байєсовська мережа складається зі схожих частин, була розглянута одна з них. Була побудована мережа для спостереження вузла IP мережі в інформаційній системі для виявлення вторгнень. Сформовані таблиці ймовірностей для вузлів графа. За допомогою спеціального програмного забезпечення була виконана візуалізація роботи математичної моделі в використанням розрахованих ймовірностей.

ВИСНОВКИ

У дипломній дослідницькій роботі за результатами виконаних теоретичних та практичних досліджень розроблено математичну модель на базі мережі Байєса.

У першому розділі були розглянуті такі основні поняття як: теорема Байєса, мережі Байєса, зловмисне програмне забезпечення та аномалії в інформаційних системах. Також було розглянуто для яких задач можуть бути використані мережі Байєса, їх класифікація і види та їх особливості. Були проаналізовані існуючі класифікації зловмисного програмного забезпечення та аномалій в інформаційних система, алгоритми машинного навчання та способи виявлення вторгнень.

У другому розділі було розглянуто і детально описано один з алгоритмів навчання математичної моделі. Були розглянуті відомі мережеві аномалії, причини та області їх виникнення в мережі, а також їх вплив на роботу інформаційних систем. Був складений перелік параметрів інформаційної системи для спостереження в системі виявлення зловмисного програмного забезпечення з метою виявлення вторгнень. Також був розглянутий приклад побудови найпростішої Байєсовської мережі та особливості застосування алгоритму ОМД для навчання математичної моделі.

У третьому розділі було сформовано загальній вигляд Байєсовської мережі для використання в системі виявлення зловмисного програмного забезпечення. Так як сформована Байєсовська мережа складається зі схожих частин, була розглянута одна з них. Була побудована мережа для спостереження вузла IP мережі в інформаційній системі для виявлення вторгнень. Сформовані таблиці ймовірностей для вузлів графа. За допомогою спеціального програмного забезпечення була виконана візуалізація роботи математичної моделі в використанням розрахованих ймовірностей.

Практична значимість отриманих результатів полягає у можливості вдосконалення методів виявлення зловмисного програмного забезпечення в інформаційних система.

За темою дипломної роботи опублікована одна стаття у фаховому науковому виданні «Збірник наукових праць Конференції АПКН-2020».

ПЕРЕЛІК ПОСИЛАННЬ

1. Что такое кибератака? [Електронний ресурс] // Официальный веб-сайт Cisco. URL: https://www.cisco.com/c/ru_ru/products/security/common-cyberattacks.html#~definition

2. Вредоносное ПО [Електронний ресурс] // Официальный веб-сайт Malwarebytes. URL: https://ru.malwarebytes.com/malware/

3. Петренко А. Разбираем EM-algorithm на маленькие кирпичики [Електронний ресурс] / А. Петренко // IT журнал Харб. 2020. URL: https://habr.com/ru/post/501850/

4. Monappa K. A. Learning Malware Analysis / Published by Packt Publishing Ltd., 2018. – с. 6-10.

5. Угрозы информационной безопасности [Електронний ресурс] // Официальный веб-сайт SearchInform (Information security). URL: https://searchinform.ru/

6. Троянские программы (Trojans) [Електронний ресурс] // Официальный веб-сайт Anti-Malware. URL: https://www.anti-malware.ru/threats/trojans

7. Вирусы-шифровальщики (Virus-Encoder, Trojan-Encoder) [Електронний ресурс] // Официальный веб-сайт Anti-Malware. URL: https://www.anti-malware.ru/threats/virus-encoder

8. Распределенные сетевые атаки / DDoS [Електронний ресурс] // Официальный веб-сайт Kaspersky. URL: https://www.kaspersky.ru/resource-center/threats/ddos-attacks

9. Рекламное ПО [Електронний ресурс] // Официальный веб-сайт Malwarebytes. URL: https://ru.malwarebytes.com/adware/

10. Ботнеты и их типы: что известно в 2018 году [Електронний ресурс] // IT журнал Харб. 2018. URL: https://habr.com/ru/post/432770/

11. Руткиты (Rootkits) [Електронний ресурс] // Официальный веб-сайт Anti-Malware. URL: https://www.anti-malware.ru/threats/rootkits

12: Руткит [Електронний ресурс] // Официальный веб-сайт ITGlobal. URL: https://itglobal.com/ru-ru/company/glossary/rootkit/

13. Лысенко А.В., Кожевникова И. С., Ананьин Е.В., Никишова А.В. Анализ методов обнаружения вредоносных программ. //Молодой учёный Международный научный журнал № 21 (125) / 2016 – с.759

14. Понимаем теорему Байеса [Електронний ресурс] // IT журнал Харб. 2019. URL: https://habr.com/ru/company/otus/blog/473468/

15. Формула полной вероятности и формулы Байеса [Електронний ресурс] // Веб-сайт Высшая математика – просто и доступно. URL: http://mathprofi.ru/formula_polnoj_verojatnosti_formuly_bajesa.html

16. Н.И. Самойленко, А.И. Кузнецов, А.Б. Костенко Теория вероятностей // Издательство «НТМТ», Харьков – 2009. с. 53 – 54

17. Байесовские сети при помощи Питона — что и зачем? [Електронний ресурс] // IT журнал Харб. 2019. URL: https://habr.com/ru/post/510526/

18. С. Николенко, А. Кадурин, Е. Архангельская Глубокое обучение. Погружение в мир нейронных сетей. // Издательство «Питер», 2018 – С.17-19.

19. Катрина Уэйкфилд Гид: алгоритмы машинного обучения и их типы. Каковы типы алгоритмов машинного обучения и когда их использовать [Електронний ресурс] // офіційний сайт SAS. URL: https://www.sas.com/ru_ru/insights/articles/analytics/machine-learning-algorithms-guide.html

20. Сканер вирусов изнутри [Електронний ресурс] // IT журнал Харб. 2012. URL: https://habr.com/ru/post/145948/

21. Введение в машинное обучение [Електронний ресурс] // IT журнал Харб 2018 URL: https://habr.com/ru/post/427867/

22. Машинное обучение для людей. Разбираемся простыми словами. [Електронний ресурс] // Авторський блог про світ технологій Вастрик.ру 2018 URL: https://vas3k.ru/blog/machine_learning/

23. Шевцова А. В. БАЄСОВСЬКА МЕРЕЖА І СИСТЕМА ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ДОСЛІДЖЕННЯ АНОМАЛІЙ / А. В. Шевцова, Т.М. Кисіль // Збірник наукових праць за матеріалами XII всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2020»,9-10 листопада, Хмельницький – 2020. – С.354-356

ДОДАТОК А

(обов’язковий)

Візуалізація роботи математичної моделі

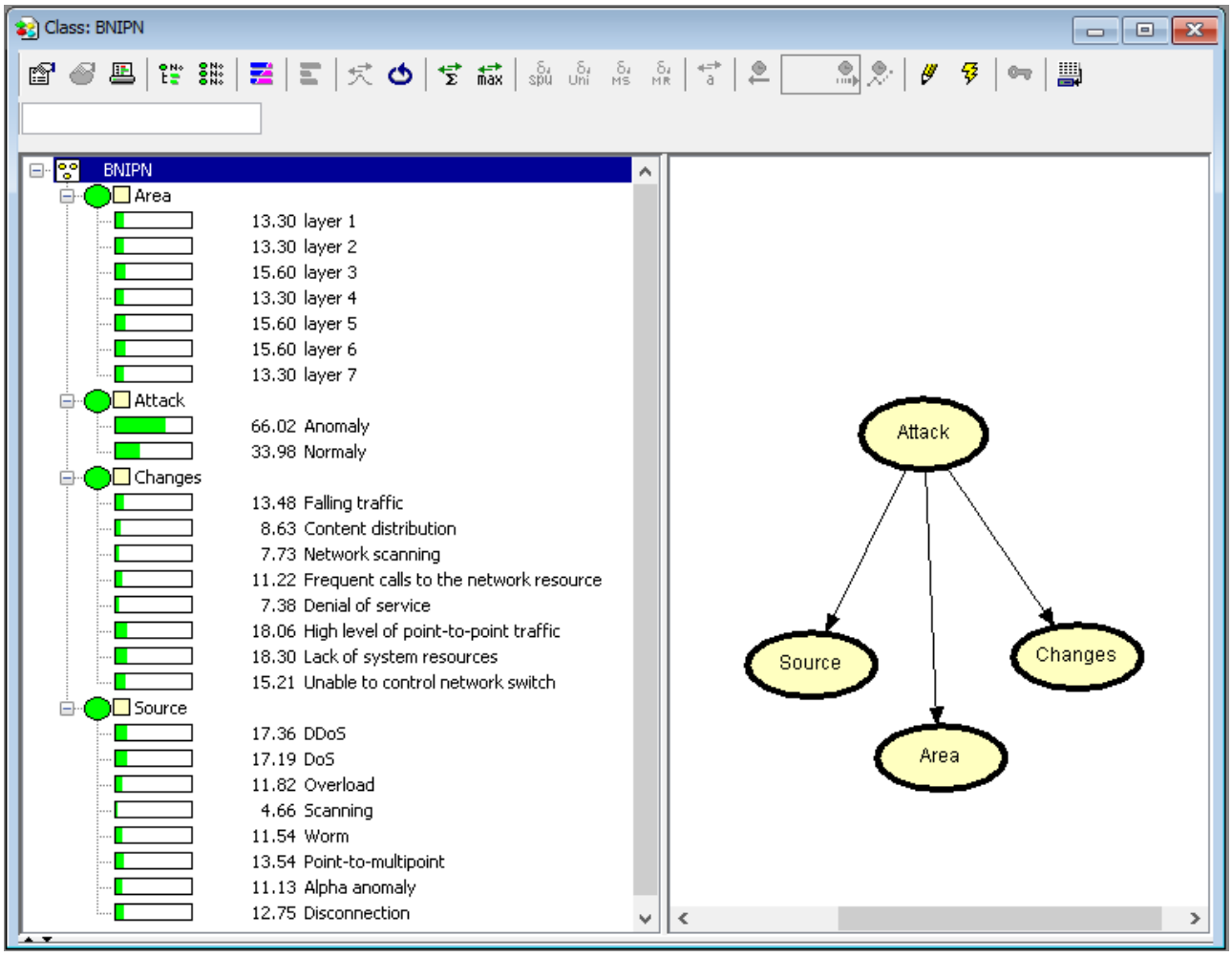


Рисунок А.1 - Візуалізація роботи математичної моделі в програмі Hugin Lite

ДОДАТОК Б

(обов’язковий)

Тези до дипломної роботи

Міністерство освіти і науки України
Хмельницький національний університет

ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XII всеукраїнської науково-практичної конференції
«Актуальні проблеми комп’ютерних наук АПКН-2020»

9-10 листопада 2020

Хмельницький 2020

УДК 004.4

Шевцова А. В., Кисіль Т. М.

Хмельницький національний університет

БАЄСОВСЬКА МЕРЕЖА І СИСТЕМА ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ДОСЛІДЖЕННЯ АНОМАЛІЙ

Розглянуто аспекти побудови математичної моделі на базі Байєсовської мережі для виявлення зловмисного програмного забезпечення в інформаційних системах, а також візуалізація роботи моделі. Представлена математична модель забезпечує виявлення зловмисного програмного забезпечення, використовуючи базу даних аномалій як вхідні дані.

Aspects of construction of mathematical model on the basis of the Bayesian network for detection of malicious software in information systems, and also visualization of the work of model are considered. The presented mathematical model provides the detection of malicious software, using the database of anomalies as input data.

На сьогоднішній день комп'ютерні та інтернет-технології розвиваються з великою швидкістю. Це змінило багато сфер діяльності людини, але також спровокувало зростання кіберзлочинності. Зловмисники використовують різноманітне зловмисне програмне забезпечення (ПЗ) для взлому інформаційних систем, використовуючи їх уразливості. До встановлення зловмисного ПЗ зазвичай призводить перехід по небезпечним посиланням або відкриття файлів, прикріплених до листа в електронній пошті. Подібні атаки завдають величезної шкоди компаніям. Типів зловмисних програм дуже багато, як і об'єктів, здатних привести до їх установлення. Їх різноманітність призводить до того, що далеко не всі зловмисні програми ми здатні виявити вже існуючими методами і програмним забезпеченням.

Мета даної дослідницької роботи це визначити, як можна використовувати Байєсовську мережу для виявлення зловмисного ПЗ в інформаційній системі.

Визначені задачі: побудувати математичну модель, провести навчання моделі, використовуючи вже існуючі алгоритми, а також візуалізувати та перевірити роботу математичної моделі при різних умовах і вхідних даних.

В основі Байєсовської мережі лежить теорема (формула) Байеса [1]:

$$P(A|B) = P(B|A)\,P(A)/P(B), \quad (1)$$

де:

P (A) – апріорна ймовірність гіпотези *A*;

P (A|B) – ймовірність гіпотези *A* при настанні події *B* (апостеріорна ймовірність);

P (B|A) – ймовірність настання події *B* при істинності гіпотези *A*;

P (B) – повна ймовірність настання події *B*.

Байєсовська мережа дозволяє відповісти на такі запроси, як:

- повідомити, що в вузлі x відбулась подія,
- визначити ймовірності події при спостережуваних аномаліях,
- визначити ймовірності причини при спостережуваних наслідках,
- визначити ймовірності однієї з причин появи події,
- провести обчислення найбільш імовірної причини спостережуваної події.

Використання Байєсовської мережі має такі переваги:

- простота побудови моделі,
- можливість роботи з нечіткими або частково відомими даними,
- можливість навчити модель в процесі її роботи.

Вхідними даними для представленої мережі виступають ідентифікатори аномалій з статистичної бази даних можливих аномалій в інформаційних системах, а вихідними даними виступають ймовірності появи події, ймовірності причин появи події, повідомлення про появу події. Вершини графа є випадковими величинами, які приймають два значення: 1 або 0, що сигналізує про наявність аномалії чи її відсутність відповідно. Дуги є ймовірнісними залежностями між величинами, значення яких задається за допомогою умовних ймовірностей.

Для побудови математичної моделі необхідно зібрати дані, створити базу даних, згенерувати вузли та дуги, визначити апріорні ймовірності і оптимізувати мережу, побудувати (навчити) мережу. Статистична база даних включає в себе аномалії в інформаційній системі, що сигналізують про наявність в ній зловмисного ПЗ. Для навчання математичної моделі був використаний EM-алгоритм, за допомогою якого проводиться оцінка параметрів моделі, на підставі якої розраховується гіпотетична ймовірність появи того або іншого результату. Кожна ітерація алгоритму включає два кроки: E (expectation) та M (maximization). На першому кроці E вибираються приховані змінні, розраховується очікуване значення функції правдоподібності. У наступних кроках E використовуються приховані змінні, які були визначені на M-кроках. На M-кроці перераховуються приховані змінні відповідно до отриманих значеннями ймовірностей на E кроці. В основі E кроку лежить формула Байєса. Алгоритм виконується до збіжності. Можна виділити наступні переваги EM-алгоритму:

- потужна статистична основа,
- при зростанні обсягу даних складність збільшується лінійно,
- алгоритм стійкий до перешкод і перепусткам даних,
- швидка збіжність при вдалій ініціалізації.

Висновки: в ході дослідницької роботи була побудована і протестована математична модель на базі Байєсовської мережі для виявлення зловмисного ПЗ з використанням різних вхідних даних, використовуючи статистичну базу даних аномалій в інформаційній системі, що наглядно демонструє ефективність використання Байєсовської мережі для вирішення проблеми кібератак.

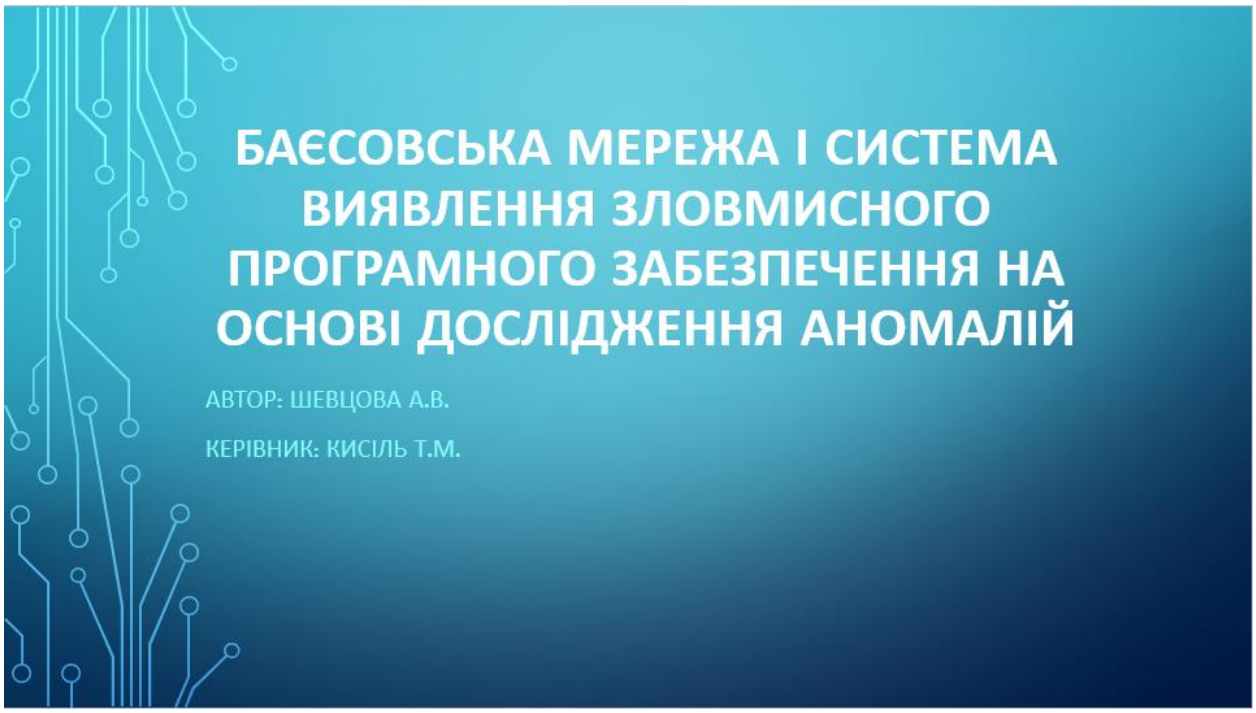
Перелік посилань:

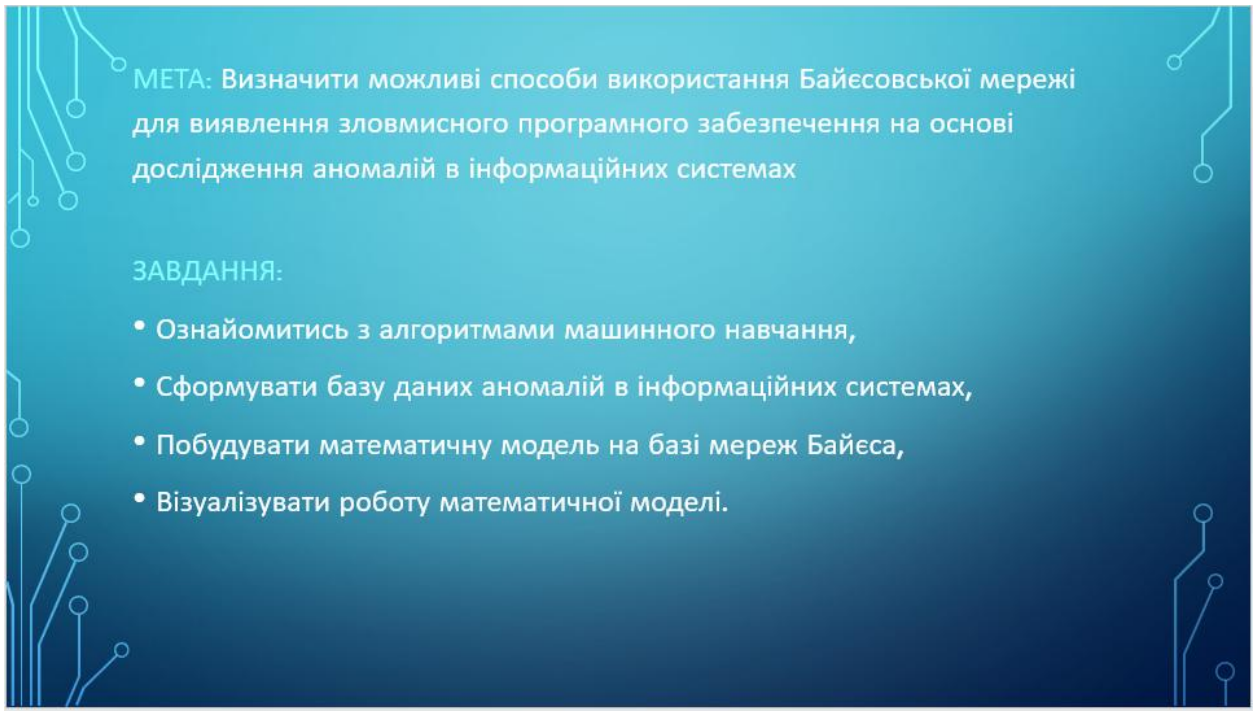
1. Что такое кибератака [Електронний ресурс] // Официальный веб-сайт Cisco. URL: https://www.cisco.com/c/ru_ru/products/security/common-cyberattacks.html
2. Вредоносное ПО [Електронний ресурс] // Официальный веб-сайт Malwarebytes. URL: https://ru.malwarebytes.com/malware/
3. Петренко А. Разбираем EM-algorithm на маленькие кирпичики [Електронний ресурс] / А. Петренко // IT журнал Харб. 2020. URL: https://habr.com/ru/post/501850/

ДОДАТОК В

(обов’язковий)

Презентація





НАУКОВА НОВИЗНА: Мережі Байєса активно використовуються для моделювання в біоінформатиці, медицині, класифікації документів, обробці зображень, обробці даних, машинному навчанні і системах підтримки прийняття рішень.

ПРАКТИЧНА ЗНАЧИМІСТЬ: Можливість вдосконалення методів виявлення зловмисного програмного забезпечення в інформаційних система.

ТЕЗИ:

Шевцова А. В. БАЄСОВСЬКА МЕРЕЖА І СИСТЕМА ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ДОСЛІДЖЕННЯ АНОМАЛІЙ / А. В. Шевцова, Т.М. Кисіль // Збірник наукових праць за матеріалами XII всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2020»,9-10 листопада, Хмельницький – 2020. – С.354-356

ДОСЛІДЖЕННЯ АНОМАЛІЙ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Методи дослідження аномалій:

- Сканування
- Евристичний аналіз
- Виявлення змін

СУЧАСНІ СИСТЕМИ ВИЯВЛЕННЯ АТАК

Класифікувати СВА можна:

- за способом реагування;
- за способом виявлення атаки;
- за способом збору інформації про атаку.

ТЕОРЕМА БАЙЄСА

$P(A|B) = P(B|A)\,P(A)/P(B)$,

де

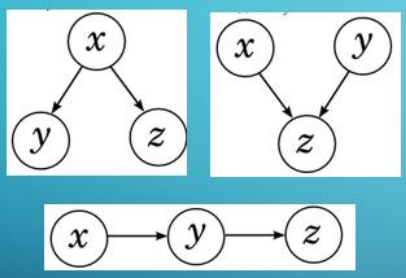
$P(A \mid B)$ - ймовірність настання події А, за умови, що відбулася подія В (апостеріорна ймовірність або умовна ймовірність);

$P(B \mid A)$ - ймовірність настання події В, за умови, що відбулася подія А (апостеріорна ймовірність);

$P(A)$ - повна ймовірність настання події А (апріорна ймовірність);

$P(B)$ - повна ймовірність настання події В (апріорна ймовірність).

МЕРЕЖІ БАЙЄСА



Зв'язок між вузлами в БМ:

- Розбіжний,
- Збіжний,
- Послідовний.

АЛГОРИТМИ МАШИННОГО НАВЧАННЯ

БУЛИ РОЗГЛЯНУТІ ТАКІ АЛГОРИТМИ МАШИННОГО НАВЧАННЯ:

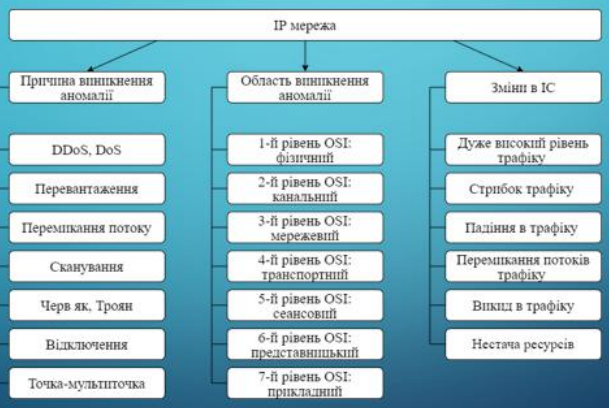
- Наївний Байєсовський класифікатор
- Алгоритм k-середніх
- Метод опорних векторів
- Лінійна регресія
- Логістична регресія
- Штучна нейронна мережа
- Дерево рішень
- Випадковий ліс
- Метод k-найближчих сусідів
- ЕМ-алгоритм
- ОМД (принцип опису мінімальної довжини)

ОСНОВНІ ПРИНЦИПИ НАВЧАННЯ МАТЕМАТИЧНОЇ МОДЕЛІ

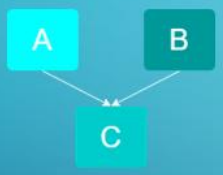
Навчання відбувається в 3 етапи:

- Аналіз даних,
- Знаходження шаблонів,
- Передбачення на основі шаблону.

РОБОТА З ВХІДНИМИ ДАНИМИ



ПРИНЦИП ПОБУДОВИ МАТЕМАТИЧНОЇ МОДЕЛІ



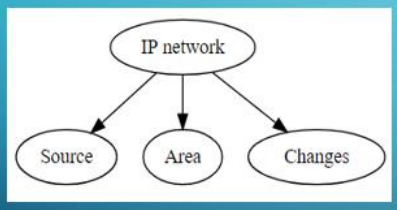
Вузол	Стан 1	Стан 2
A	1	0
B	1	0
C	1	0

Апріорна ймовірність p (A)	
A = 1	A = 0
0.1	0.9

Апріорна ймовірність p (B)	
B = 1	B = 0
0.1	0.9

Таблиця умовних ймовірностей p (C | A, B)				
	B = 1		B = 0	
	A = 1	A = 0	A = 1	A = 0
C = 1	0.95	0.85	0.90	0.02
C = 0	0.5	0.15	0.10	0.98

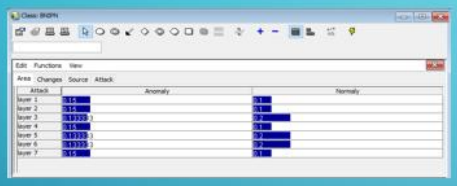
БМ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЇ В МЕРЕЖІ IP

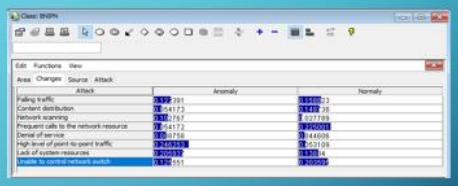


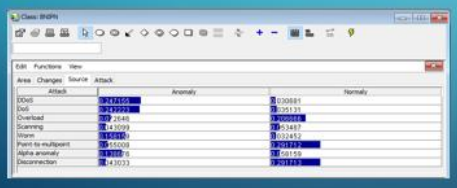
Де

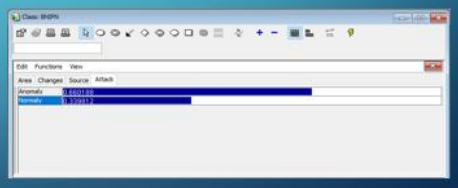
- «IP network» - вузол, що перевіряється,
- «Source» - причина виникнення аномалії,
- «Area» - область виникнення аномалії,
- «Changes» – зміни в IC.

БМ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЇ В МЕРЕЖІ IP

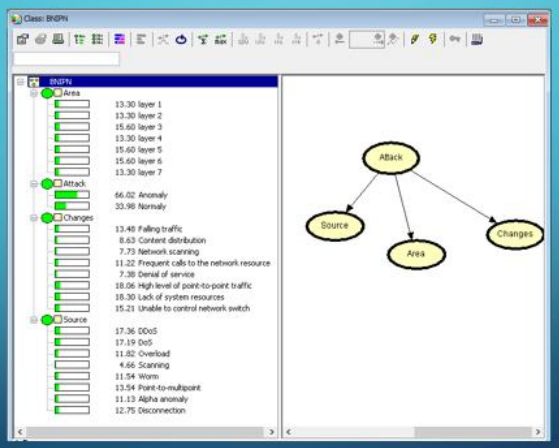


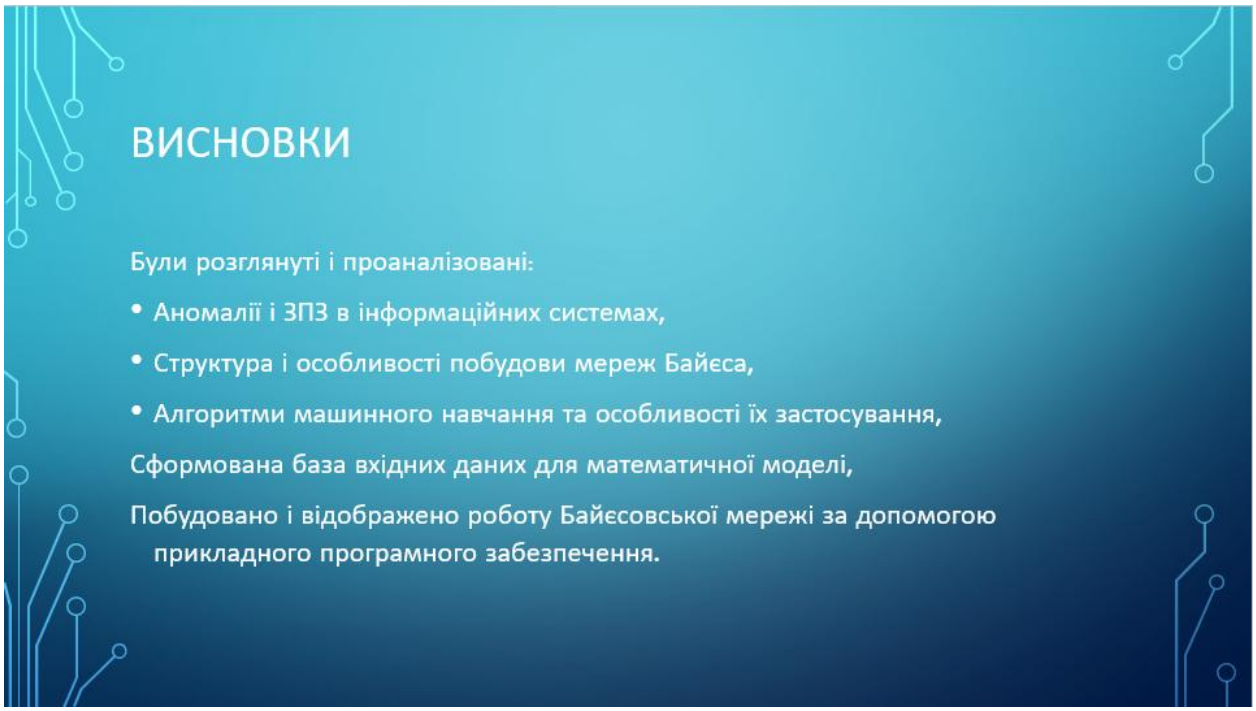






БМ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЇ В МЕРЕЖІ IP




ВИСНОВКИ
Були розглянуті і проаналізовані:
• Аномалії і ЗПЗ в інформаційних системах,
• Структура і особливості побудови мереж Байєса,
• Алгоритми машинного навчання та особливості їх застосування,
Сформована база вхідних даних для математичної моделі,
Побудовано і відображено роботу Байєсовської мережі за допомогою прикладного програмного забезпечення.

UNICHECK

Имя пользователя:
Kafedra TMIT KhNU

ID проверки:
1005321431

Дата проверки:
01.12.2020 21:08:45 EET

Тип проверки:
Doc vs Internet + Library

Дата отчета:
01.12.2020 21:20:11 EET

ID пользователя:
100005657

Название файла: Шевцова ПМ-19-1

Количество страниц: 75 Количество слов: 10756 Количество символов: 79450 Размер файла: 1.27 MB ID файла: 1005444322

3.62%
Совпадения

Наибольшее совпадение: 0.77% с Интернет-источником (https://er.nau.edu.ua/bitstream/NAU/35921/1/%D0%A1%D0%...

2.95% Источники из Интернета 221 ..... Страница 77

1.06% Источники из Библиотеки 30 ..... Страница 78

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

0% Исключений

Нет исключенных источников

Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 14

Mon Nov 30 15:50:25 EET 2020, Стецюк Віктор Іванович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 2.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 8%**

ID: 81722 Название: Баєсовська мережа і система виявлення зловмисного програмного забезпечення на основі дослідження аномалій Добавлено в БД: 2020-11-30 Авторы: Шевцова Анастасія Володимирівна Руководители: Кисіль Тетяна Миколаївна Консультанты: Оппоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	73625	710	2475 (3%)	35 (5%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник студентка групи ПМм-19-1 Шевцова А.В.

Тема Байєсовська мережа і система виявлення зловмисного програмного забезпечення на основі дослідження аномалій

Спеціальність 113 – Прикладна математика

Обсяг дипломної роботи:

Кількість листів креслень 0; кількість сторінок записки

1.Короткий зміст ДР та прийнятих рішень Представлена робота присвячена актуальній темі визначення можливих способів використання Байєсовської мережі для виявлення зловмисного програмного забезпечення на основі дослідження аномалій в інформаційних систем і складається з наступних розділів: вступ, аналіз відомих моделей, методів та засобів, проектування програмного забезпечення для вирішення проблеми, реалізація і візуалізація мережі, висновки, додатки.

2. Висновок про відповідність ДР поставленому завданню Магістерська кваліфікаційна робота виконана у відповідності з завданням із дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі студентка провела детальний аналіз предметної області, дала визначення баєсовської мережі та описала аномалії, можливі в інформаційних системах. В другому розділі здійснено аналіз методів машинного навчання. В третьому розділі аргументовано використання саме Баєсовських мереж для виявлення зловмисного програмного забезпечення. В четвертому розділі побудовано та візуалізовано таку модель за допомогою прикладного програмного забезпечення Hugin Lite

4. Позитивні сторони роботи До позитивних сторін роботи слід віднести актуальність даного направлення дослідження, деталізацію аналізу усіх розглянутих стратегій вирішення проблеми та поглиблене опрацювання всіх аспектів реалізації з практичним використанням запропонованого рішення.

5. Негативні сторони роботи __ До негативних сторін роботи слід віднести недоліки по оформленню представленого матеріалу, що були виправлені.

6. Оцінка графічного оформлення та пояснювальної записки роботи _Дані матеріали роботи є структурованими у чіткій та логічній формі та відображають послідовність виконання поставлених завдань. І хоча й в них було знайдено декілька стилістичних та орфографічних помилок, вони були пізніше усунені. Тому дане виконання пояснювальної записки та графічного оформлення заслуговує оцінки «добре».

7. Відгук про роботу в цілому ___Загалом, зміст представленої роботи в повній мірі розкриває обрану тему. Дослідження, проведені в матеріалах є достатньо аргументованими. Прослідковуються високі теоретичні та практичні рівні у даному виконанні. Результатом проведення досліджень стали відповідні висновки і конкретні пропозиції щодо вдосконалення процесу виявлення зловмисного програмного забезпечення із використанням Байєсовської мережі.

8. Інші зауваження

9. Оцінка дипломної роботи ___Робота заслуговує оцінки «добре», а її автор – присвоєння кваліфікації «магістра» з прикладної математики.

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи) Лисенко Сергій Миколайович, доктор технічних наук, доцент кафедри комп'ютерної інженерії та системного програмування ХНУ

"____" _____ 2020 р. _____
(підпис)

Завідувачу кафедри ТМІТ
д-р.техн.наук Підченку С.К.

Шевцова А.В.
ПІБ здобувача вищої освіти

ФПКТС, 2 курсу, групи ПМм-19-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

01.12.2002
дата

підпис

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Баєсовська мережа і система виявлення зловмисного програмного забезпечення на основі дослідження аномалій

Автор: Шевцова Анастасія Володимирівна

Спеціальність: 113 – прикладна математика

Освітня програма: освітньо-професійна

Науковий керівник: Кисіль Тетяна Миколаївна, к.ф.-м.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	+
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) співпадіння розміщені в розділах, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення мають фрагментарний характер, або мають належним чином оформленні посилання;
3) більшість джерел запозичення дублюють одне одного.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 3,62% і адресується до 221 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Дата

Підченко С.К.
Підпис

Кисіль Т.М.
Підпис