Вінницький національний технічний університет
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

ПОЛЬГУЛЬ ТЕТЯНА ДМИТРІВНА

УДК 004.8:044.89

ДИСЕРТАЦІЯ

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВИЯВЛЕННЯ ШАХРАЙСТВА ПРИ ІНСТАЛЮВАННІ МОБІЛЬНИХ ДОДАТКІВ З ВИКОРИСТАННЯМ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

05.13.06 – інформаційні технології
Технічні науки

Подається на здобуття наукового ступеня кандидата технічних наук

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

Науковий керівник Яровий Андрій Анатолійович
доктор технічних наук, професор

Вінниця – 2020

АНОТАЦІЯ

Польгуль Т. Д. Інформаційна технологія виявлення шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 «Інформаційні технології». – Вінницький національний технічний університет, Вінниця, 2020.

У зв'язку з появою на ринку значної кількості мобільних додатків, якими користуються мільярди користувачів, компанії-розробники мобільних додатків звертаються за послугами маркетингових кампаній з метою залучення інсталювань саме до їхнього додатку. Саме така потреба у маркетингових кампаніях стала однією з причин появи шахраїв та їх шахрайських способів інсталювання мобільних додатків. Шахраї, у свою чергу, приводять до компаній-розробників необхідну кількість фейкових (несправжніх) «користувачів» та отримують за це відповідну грошову винагороду. Проте такі «користувачі» ніколи не повертаються у мобільний додаток, оскільки є фейковими, ми ж їх називатимемо шахрайськими.

У наш час вже існують такі відомі види шахрайства при інсталюванні додатків, як мобільне викрадення (mobile hijacking), кліковий спам (click spamming), ферми дій (action farms), а також методи та системи виявлення шахрайства при інсталюванні мобільних додатків такі, як Fraudlogix та Kraken, Adjust, Kochava та TCM Attribution Analytics, Protect360 від Appsflyer, FraudScore та AppMetrica. Проте необхідно зазначити, що лише останні дві використовують інтелектуальну складову, причому система AppMetrica просто ґрунтується на алгоритмах та API системи FraudScore. Але вказані системи-аналоги виконують рейтингування користувачів на основі не всіх, а вибіркових вхідних даних, тому відбувається упущення шахраїв системами. Інші вказані системи використовують існуючі бази з шахрайськими даними (наприклад, IP-адресами),

що також призводить до упущення шахраїв, які мають інші властивості, шаблони, поведінку.

Очевидно, що причиною вищевказаних недоліків систем є відсутність єдиного підходу до виявлення шахрайства на основі всіх наявних даних. Також, недоліком існуючих систем є те, що вони розпізнають лише відомі види шахрайства і не можуть розпізнавати нові шахрайські шаблони. А в сучасному світі важливою є можливість системи адаптуватись, тому необхідним є створення відповідних інформаційних технологій, що матимуть змогу самонавчатися.

Все вищенаведене є передумовою актуальності створення інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків, яка б відстежувала та визначала шаблони шахраїв, які непомітні людині. Розв’язанню цієї задачі присвячена дана робота.

Метою роботи дисертаційного дослідження є підвищення точності та швидкодії процесів виявлення шахрайства при інсталюванні мобільних додатків.

Для досягнення вказаної мети в роботі розв’язуються такі основні задачі:

– аналіз методів та постановка задачі виявлення шахрайства при інсталюванні мобільних додатків;

– формалізація процесу виявлення шахрайства як аномалії в даних;

– аналіз та класифікація різнорідних даних при інсталюванні мобільних додатків;

– розробка узагальненого методу виявлення шахрайства при інсталюванні мобільних додатків;

– розробка методу подолання різнорідності вхідних даних;

– розробка інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків.

Науковою новизною виконаного дисертаційного дослідження визначено:

1. Вперше запропоновано метод подолання різнорідності вхідних даних, що являє собою сукупність процедур вибору ознак, зниження розмірності та

нормалізації даних, відмінність якого полягає у новій моделі процесу подолання різнорідності даних шляхом шкалювання за інформативністю, що дозволяє всю множину різнорідних даних про користувачів звести до вектору уніфікованих ознак без зменшення діагностичної цінності інформації.

2. Удосконалено модель класифікації користувачів на основі глибинних нейронних мереж у частині зниження розмірності та нормалізації даних згідно запропонованого методу подолання різнорідності даних, яка є основою для створення узагальненого портрету шахрая з метою спрощення процесів їх виявлення.

3. Вперше розроблено узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків, відмінність якого полягає у використанні запропонованої моделі класифікації користувачів та методу подолання різнорідності вхідних даних, що дозволяє визначити класи користувачів та підвищити точність виявлення шахрайства при інсталюванні мобільних додатків.

Практична цінність отриманих в дисертації результатів полягає у наступному: здійснено класифікацію різнорідних даних, що дозволило спростити процес аналізу різнорідних за метриками, розмірностями і шаблонами даних та автоматизувати його; розроблено алгоритм пошуку аномалій в даних, алгоритми процесу подолання різнорідності вхідних даних, алгоритм виявлення шахрая при інсталюванні мобільних додатків, алгоритм створення узагальненого портрету шахрая та алгоритм мінімізації часу виявлення шахраїв на основі розпаралелення обчислювальних процесів, які покладені в основу інформаційної технології, що підвищило точність та швидкодію виявлення шахраїв; запропоновано інформаційну технологію виявлення шахрайства при інсталюванні мобільних додатків, яка використовує запропоновані метод виявлення шахрайства при інсталюванні мобільних додатків, метод подолання різнорідності вхідних даних, модель класифікації даних, і, на відміну від існуючих систем Antifraud, дозволила підвищити точність класифікації

користувачів до 99,14 %, зокрема точність класифікації шахраїв – до 82,76 %; розроблено програмне забезпечення “Mobile App Install Fraud Detection System” для виявлення шахрайства при інсталюванні мобільних додатків.

Результати дисертаційної роботи доповідались та обговорювались на 9 науково-технічних конференціях: XLV, XLVI, XLVIII науково-технічних конференціях професорсько-викладацького складу, співробітників та студентів ВНТУ (2016, 2017, 2019 рр.); науково-практичній конференції «Сучасні тенденції розвитку системного програмування» (м. Київ, Національний авіаційний університет, 2016 р.); XIV Міжнародній конференції «Контроль і управління в складних системах (КУСС-2018)» (м. Вінниця, ВНТУ, 2018 р.); V Міжнародній науково-технічній конференції студентів, магістрів та аспірантів «Інформатика, управління та штучний інтелект» (м. Харків, Національний технічний університет «Харківський політехнічний інститут», 2018 р.); 5th International Winter School on Big Data BigDat2019 (м. Кембридж, University of Cambridge, Великобританія, 2019 р.); 577th International Conference on Innovative Engineering Technologies (ICIET) (м. Бангкок, Таїланд, 2019 р.); The 10th International Conference on Dependable Systems, Services and Technologies (DESSERT’2019) (м. Лідс, Великобританія, Leeds Beckett University, 2019 р.), де отримано нагороду «Best Paper Award»; The 14th International conference "Computer sciences and Information technologies" (CSIT 2019) (м. Львів, Україна, 2019 р.); The 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), (м. Метц, Франція, 2019 р.).

Результати дисертаційної роботи впроваджені на підприємствах та навчальних закладах: Garuda AI B. V. (м. Схіпгол, Нідерланди) – інформаційна технологія; ТОВ «ВІН ІНТЕРАКТИВ» (м. Вінниця, Україна) – алгоритми подолання різнорідності даних, модель процесу подолання різнорідності даних; ТОВ «4ХайТек» (м. Вінниця, Україна) – модель процесу подолання різнорідності вхідних даних, методика виявлення шахрая при інсталюванні

мобільних додатків; ПП «Літсофт» (м. Київ, Україна) – алгоритм подолання різнорідності даних, узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків, методика створення узагальненого портрету шахрая; у навчальний процес кафедри комп'ютерних наук Вінницького національного технічного університету (ВНТУ) – узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків та у навчальний процес кафедри інформатики, програмної інженерії та економічної кібернетики Херсонського державного університету – інформаційна технологія виявлення шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних.

Ключові слова: інформаційна технологія, інтелектуальний аналіз даних, виявлення шахрайства, виявлення аномалій в даних, коефіцієнти подібності, модель класифікації, машинне навчання, глибинні нейронні мережі, метод подолання різнорідності, матриця невідповідності, інсталювання мобільних додатків.

ABSTRACT

Polhul T. D. Information technology for fraud detection during mobile applications installation using data mining. – Qualification research paper, manuscript copyright.

Thesis for the degree of a candidate of technical sciences in specialty 05.13.06 «Information technology». – Vinnytsia National Technical University, Vinnytsia, 2020.

Nowadays there appeared a need to create information technology for fraud detection. This is due to the emergence of a huge number of new competing products among the billions of users in the mobile application market. To attract most users to their apps, mobile apps developers use the marketing campaigns service. Such a need in marketing campaigns has led to the massive appearance of fraudsters and fraudulent types of mobile applications installation, which can bring companies the required number of "users" and receive appropriate money reward for it [1]. However, it should be noted that such "users" never return to the mobile application because they are fake, we will call them fraudulent ones. Therefore, the creation of information technology for fraud detection during mobile applications installation is an important task.

For the time being, there are already known types of mobile applications installation fraud such as mobile hijacking, click spamming, action farms [2-7] and methods and systems for fraud detection during mobile applications installation such as: Fraudlogix and Kraken, Adjust, Kochava and TCM Attribution Analytics, Protect360 by Appsflyer, FraudScore and AppMetrica mentioned in the research paper. However, it should be noted that only the last couple uses the intelligent component, where AppMetrica simply relying on FraudScore and using its algorithms and API. But even the above-mentioned systems perform user rating based on not all but selective input data, so there is an omission of fraudsters by the system. Other specified systems use existing databases with fraudulent data (for example, IP-addresses), which

also leads to the omission of fraudsters having other properties, patterns, behavior, by such systems.

Obviously, the reason for the above mentioned disadvantages of the systems is the lack of a single approach for fraud detection based on available data. Also, the disadvantage of existing systems is that they recognize only known types of fraud and cannot recognize new fraudulent patterns. And the ability of the system to adapt is important in the modern world, therefore, it is necessary to create an intellectual information technology that will be able to learn.

All of the above is a prerequisite to creating an information technology for fraud detection during mobile applications installation that would track and detect fraudulent patterns unnoticed by humans. This research paper is dedicated to solve this problem.

The purpose of the qualification research paper is to improve the accuracy and speed of fraud detection process during mobile applications installation.

In order to achieve the mentioned purpose, the following basic tasks are solved in the work:

– analysis of methods and statement of fraud detection during mobile applications installation problem;

– formalization of the process of detection of fraud as an anomaly in the data;

– analysis and classification of heterogeneous data during mobile applications installation;

– development of a generalized method for fraud detection during mobile applications installation;

– development of a method of overcoming heterogeneity of input data;

– developing information technology for fraud detection during mobile applications installation.

The scientific novelty of the qualification research paper is:

1. For the first time, a method of overcoming the heterogeneity of input data is proposed, which is a set of procedures for feature selecting, dimensionality reduction and data normalization, the difference of which lies in the new model of overcoming

the heterogeneity of data by scaling information, which allows the whole set of heterogeneous user data to be reduced to a vector, reducing the diagnostic value of information.

2. The model of users' classification based on deep neural networks in terms of dimensionality reduction and data normalization has been improved according to the proposed method of overcoming heterogeneity of data, which is the basis for creating of general fraudsters fingerprint in order to simplify their detection processes.

3. For the first time, a generalized method for fraud detection in during installation of mobile applications has been developed, the difference being the use of the proposed users' classification model and the method of overcoming the heterogeneity of the input data, which allows defining users’ classes and increasing the reliability of fraud detection during mobile app installs.

The practical value of the results obtained in the qualification research paper is as follows: classification of heterogeneous data was carried out, which made it possible to simplify the process of analysis of data which is heterogeneous by metrics, dimensions and data templates and to automate it; an algorithm for detecting anomalies in data, algorithms for the process of overcoming the heterogeneity of input data, algorithm for fraud detection during mobile applications installations, an algorithm for generalized fraudster fingerprint formation, and algorithm for minimizing the time of fraud detection based on the parallelization of computing processes, which are the basis of information technology, which has increased the accuracy and speed of detection of fraudsters, were developed; information technology for fraud detection during mobile applications installation has been proposed for the first time, that uses the following: a generalized method for fraud detection during mobile applications installation, a method for overcoming the heterogeneity and classification model of user input data, which, unlike existing Antifraud technologies, allowed to improve accuracy of users’ classification to 99,14 %, in particular, detecting fraudulent users to 82,95 %; “Mobile App Install Fraud Detection System” software for fraud detection during mobile applications installation has been developed.

The results of the qualification research paper were reported and discussed at 9 scientific and technical conferences: XLV, XLVI, XLVIII scientific and technical conferences of teaching staff, staff and students of Vinnitsa National Technical University; scientific-practical conference "Modern tendencies of development of system programming" (Kyiv, National Aviation University, 2016); XIV International Conference "Control and Control in Complex Systems (КУСС-2018)» (Vinnitsa, Vinnitsa National Technical University, 2018); V International Scientific and Technical Conference of Students, Masters and Postgraduate Students "Informatics, Management and Artificial Intelligence" (Kharkiv, National Technical University "Kharkiv Polytechnic Institute", 2018); 5th International Winter School on Big Data BigDat2019 (Cambridge, University of Cambridge, UK, 2019); 577th International Conference on Innovative Engineering Technologies (ICIET) (Bangkok, Thailand, 2019); The 10th International Conference on Dependable Systems, Services and Technologies (DESSERT'2019) (Leeds, UK, Leeds Beckett University, 2019), where the «Best Paper Award» was recieved; The 14th International conference "Computer sciences and Information technologies" (CSIT 2019) (Lviv, Ukraine, September 17-20, 2019); The 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), (Metz, France, September 18-21, 2019).

The results of the qualification research paper were implemented at Garuda AI B.V. (Netherlands) – information technology; LLC «WinInteractive» – algorithms for overcoming data heterogeneity, model for the process of overcoming data heterogeneity; LLC «4HighTech» – a model of the process of overcoming the heterogeneity of the input data, the method of fraud detection when installing mobile applications; PE «Litsoft» – algorithm for overcoming heterogeneity of data, a generalized method of detecting fraud when installing mobile applications, a method for creating a generalized fraudster's fingerprint; to the educational process of the Computer Science Department of Vinnytsia National Technical University – a generalized method for fraud detection during mobile applications installation; and to

the educational process of the Department of Informatics, Software Engineering and Economic Cybernetics of Kherson State University – information technology for fraud detection during mobile applications installation using data mining.

Keywords: information technology, data mining, fraud detection, anomaly detection, similarity coefficients, classification model, machine learning, deep neural networks, method for overcoming heterogeneity of data, confusion matrix, mobile applications installation.

СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

[1] T. Polhul, and A. Yarovyi, “Development of a method for fraud detection in heterogeneous data during installation of mobile applications”, *Eastern-European Journal of Enterprise Technologies*, no. 1/2 (97), 2019. doi: 10.15587/1729-4061.2019.155060

[2] A. Yarovyi, and T. Polhul, “Applied Aspects of Implementation of Intelligent Information Technology for Fraud Detection During Mobile Applications Installation”, *Advances in Intelligent Systems and Computing IV. CCSIT 2019: Advances in Intelligent Systems and Computing, Springer*, Cham, Switzerland, vol 1080, pp. 377-386, 2019. doi: https://doi.org/10.1007/978-3-030-33695-0_26

[3] T. Polhul, “Conceptual Model of an Intelligent System for Detecting Fraud During Mobile Applications Installation”, in *Proc. 10th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Leeds, United Kingdom, pp. 167-174, 2019. doi: 10.1109/DESSERT.2019.8770030. Режим доступу: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8770030&isnumber=8770005.

[4] T.D. Polhul, A.A. Yarovyi, R.Romaniuk, P.Komada, N. Askarova "Method of data anomaly detection in the process of mobile applications installation", *Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019*, 111761Y; https://doi.org/10.1117/ 12.2536855

[5] T. Polhul and A. Yarovyi, "Method of Fraudster Fingerprint Formation During Mobile Application Installations", *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Metz, France, 2019, pp. 1099-1103. doi: 10.1109/IDAACS.2019.8924369. Режим доступу: https://ieeexplore.ieee.org/ document/8924369.

[6] A. Yarovyi, and T. Polhul, “Intelligent information technology for fraud detection during mobile applications installation”, in *Proc. 14th International conference "Computer sciences and Information technologies" (CSIT 2019)*, pp. 1-5, Lviv, 2019. doi: 10.1109/STC-CSIT.2019.8929827. Режим доступу: https://ieeexplore.ieee.org/ document/8929827.

[7] T. Polhul, “Development of an intelligent system for detecting mobile app install fraud”, *International Journal of Advances in Electronics and Computer Science (IJAECS)*, vol. 6, no. 7, pp. 13-17, July, 2019.

[8] А.А. Яровий, О.Н. Романюк, І.Р. Арсенюк, та Т.Д. Польгуль, “Виявлення шахрайства при інсталюванні програмних додатків з використанням інтелектуального аналізу даних”, *Наукові праці Донецького національного технічного університету. Серія “Інформатика, кібернетика та обчислювальна техніка”*, № 2 (25), с. 126-131, 2017. Режим доступу: http://science.donntu.edu.ua/ wp-content/uploads/2018/03/ikvt_2017_2_site-1.pdf

[9] Т. Д. Польгуль, та А. А. Яровий, “Метод подолання різнорідності даних для виявлення шахрайства при інсталюванні мобільних додатків”, *Вісник СНУ ім. В. Даля – Сєвєродонецьк: СНУ ім. В. Даля*, № 7 (248), с.60-69, 2018.

[10] Т.Д. Польгуль, та А.А. Яровий, “Аналіз різнорідних даних в інтелектуальних системах виявлення шахрайства”, *Вісник Вінницького політехнічного інституту*, № 2, с. 78-90, 2019.

[11] Т.Д. Польгуль, “Інформаційна технологія побудови інтелектуальних систем виявлення шахрайства при інсталюванні мобільних додатків”, *Інформаційні технології та комп'ютерна інженерія*, № 1, с. 4-16, 2019.

[12] T. Polhul, “Development of an intelligent system for detecting mobile app install fraud”, in *Proc. IRES 156th International Conference*, Bangkok, Thailand, 2019, pp. 25-29.

[13] А.А. Яровий, та Т.Д. Польгуль, “Комп'ютерна програма “Програмний модуль збору даних інформаційної технології виявлення шахрайства при інсталюванні програмних додатків”, *Свідоцтво про реєстрацію авторського права на твір № 76348*, К.: Міністерство економічного розвитку і торгівлі України, 2018.

[14] А.А. Яровий А. А., та Т.Д. Польгуль, “Комп'ютерна програма “Програмний модуль визначення схожості користувачів інформаційної технології виявлення шахрайства при інсталюванні програмних додатків”, *Свідоцтво про реєстрацію авторського права на твір № 76347*, К.: Міністерство економічного розвитку і торгівлі України, 2018.

[15] Т.Д. Польгуль, та А.А. Яровий, “Визначення шахрайських операцій при встановленні мобільних додатків з використанням інтелектуального аналізу даних”, *Сучасні тенденції розвитку системного програмування. Тези доповідей*, Київ, 2016, с. 55-56. Режим доступу: http://ccs.nau.edu.ua/wp-content/uploads/2017/12/%D0%A1%D0%A2%D0%A0%D0%A1%D0%9F_2016_07.pdf

[16] А. Яровий, Т. Польгуль, та Л. Крилик, “Розробка методу виявлення шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних”, *XIV Міжнародна конференція Контроль і управління в складних системах (КУСС-2018). Тези доповідей*, Вінниця, 2018, с. 35.

[17] А.А. Яровий, та Т.Д. Польгуль, “Подолання різнорідності вхідних даних при виявленні шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних”, на *П’ятій міжнародній науково-технічній конференції студентів, магістрів, аспірантів «Інформатика, управління та штучний інтелект»*, Національний технічний університет «Харківський політехнічний інститут», Харків, 2018, с. 109.

[18] Т.Д. Польгуль, та А.А. Яровий, “Визначення шахрайських операцій при інсталяції мобільних додатків з використанням інтелектуального аналізу даних”, на *XLVI науково-технічній конференції підрозділів ВНТУ*, Вінниця, 2017. Режим доступу: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/17200/2158.pdf?sequence=3

[19] Т.Д. Польгуль, “Моделювання процесу виявлення шахрайства при інсталюванні мобільних додатків”, на *XLVIII науково-технічній конференції підрозділів ВНТУ*, Вінниця, 2019. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2019/paper/view/6863

[20] Т.Д. Польгуль, “Порівняльний аналіз Apache Spark та Apache Flink для роботи з Big Data”, на *XLV науково-технічній конференції підрозділів ВНТУ*, Вінниця, 2016. Режим доступу: https://ir.lib.vntu.edu.ua/handle/123456789/11619

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ACID	Atomicity, consistency, isolation, durability
ANN	Artificial neural networks
API	Application programming interface
CNN	Convolutional neural network
DNN	Deep neural network
DQN	Deep Q Learning
EM	Expectation-maximization
GAN	Generative Adversarial Nets
LB	Load Balancer
LOF	Local outlier factor
MTTI	Mean-time-to-install
NN	Neural networks
RNN	Recurrent neural network
SVM	Support Vector Machine
TPR	True positive rate
TTI	Time-to-install Outliers
ІАД	Інтелектуальний аналіз даних
ІТ	Інформаційна технологія
КН	Комп’ютерні науки
ШІ	Штучний інтелект

ВСТУП

Обґрунтування вибору теми дослідження. У зв’язку з появою на ринку значної кількості мобільних додатків, якими користуються мільярди користувачів, компанії-розробники мобільних додатків звертаються за послугами маркетингових кампаній з метою залучення інсталювань саме до їхнього додатку. Саме така потреба у маркетингових кампаніях стала однією з причин появи шахраїв та їх шахрайських способів інсталювання мобільних додатків. Шахраї, у свою чергу, приводять до компаній-розробників необхідну кількість фейкових (несправжніх) «користувачів» та отримують за це відповідну грошову винагороду. Проте такі «користувачі» ніколи не повертаються у мобільний додаток, оскільки є фейковими, ми ж їх називатимемо шахрайськими.

У наш час вже існують такі відомі види шахрайства при інсталюванні додатків, як мобільне викрадення (mobile hijacking), кліковий спам (click spamming), ферми дій (action farms) [1]-[3], а також методи та системи виявлення шахрайства при інсталюванні мобільних додатків такі, як Fraudlogix [4] та Kraken [5], Adjust [6], Kochava [7] та TCM Attribution Analytics [8], Protect360 [9] від Appsflyer [10], FraudScore [11] та AppMetrica [12], які згадуються у роботі [13]. Проте необхідно зазначити, що лише останні дві використовують інтелектуальну складову, причому система AppMetrica просто ґрунтується на алгоритмах та API системи FraudScore. Але вказані системи-аналоги виконують рейтингування користувачів на основі не всіх, а вибіркових вхідних даних, тому відбувається упущення шахраїв системами [14]-[16]. Інші вказані системи використовують існуючі бази з шахрайськими даними (наприклад, IP-адресами), що також призводить до упущення шахраїв, які мають інші властивості, шаблони, поведінку.

Очевидно, що причиною вищевказаних недоліків систем є відсутність єдиного підходу до виявлення шахрайства на основі всіх наявних даних. Також, недоліком існуючих систем є те, що вони розпізнають лише відомі види шахрайства і не можуть розпізнавати нові шахрайські шаблони. А в сучасному

світі важливою є можливість системи адаптуватись, тому необхідним є створення відповідних інформаційних технологій, що матимуть змогу самонавчатися.

Все вищенаведене є передумовою актуальності створення інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків, яка б відстежувала та визначала шаблони шахраїв, які непомітні людині [16]. Розв’язанню цієї задачі присвячена дана робота.

Зв’язок роботи з науковими програмами, планами, темами. Дисертаційне дослідження проводилось згідно з планами науково-дослідних робіт кафедри комп'ютерних наук Вінницького національного технічного університету, в тому числі в межах:

- наукового проекту Ф61/199-20150/4711 “Методологія побудови високопродуктивних інтелектуалізованих паралельно-ієрархічних систем на основі сучасних мережевих обчислювальних комплексів з гетерогенною архітектурою” (№ державної реєстрації: 0115U001975, 2015 р.), при виконанні якого автор брала участь як виконавець окремих підрозділів;
- кафедральної теми 47 К2 "Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій" (2016-2018 р.), при виконанні якої автор брала участь як відповідальний виконавець,
- кафедральної теми 22 К1 "Розробка спеціалізованих засобів штучного інтелекту на основі інтелектуального аналізу даних та машинного навчання" (2019 р.); при виконанні якої автор бере участь як виконавець окремих підрозділів.

Мета і завдання дослідження.

Метою роботи є підвищення точності та швидкодії процесів виявлення шахрайства при інсталюванні мобільних додатків.

Для досягнення вказаної мети в роботі розв’язуються такі основні задачі:

- аналіз методів та постановка задачі виявлення шахрайства при інсталюванні мобільних додатків;

– формалізація процесу виявлення шахрайства як аномалії в даних;

– аналіз та класифікація різнорідних даних при інсталюванні мобільних додатків;

– розробка узагальненого методу виявлення шахрайства при інсталюванні мобільних додатків;

– розробка методу подолання різнорідності вхідних даних;

– розробка інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків.

Об'єкт дослідження – процеси виявлення шахрайства як аномалій в даних при інсталюванні мобільних додатків.

Предмет дослідження – моделі, методи та інформаційні технології виявлення шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних.

Методи дослідження, що використані в роботі: методи шкалювання під час вирішення задач аналізу та класифікації різнорідних даних при інсталюванні мобільних додатків та розробки методу подолання різнорідності вхідних даних; теорія множин для вирішення задачі формалізації процесу виявлення шахрайства як аномалії в даних, а також методи класифікації, статистичні методи, методи машинного навчання, інтелектуальний аналіз даних, методи кластеризації, нейромережеві методи для вирішення задач розробки узагальненого методу виявлення шахрайства при інсталюванні мобільних додатків та розробки інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків.

Наукова новизна отриманих результатів. В ході розв'язання поставлених задач були отримані наукові результати.

1. Вперше запропоновано метод подолання різнорідності вхідних даних, що являє собою сукупність процедур вибору ознак, зниження розмірності та нормалізації даних, відмінність якого полягає у новій моделі процесу подолання різнорідності даних шляхом шкалювання за інформативністю, що дозволяє всю

множину різнорідних даних про користувачів звести до вектору уніфікованих ознак без зменшення діагностичної цінності інформації.

2. Удосконалено модель класифікації користувачів на основі глибинних нейронних мереж у частині зниження розмірності та нормалізації даних згідно запропонованого методу подолання різнорідності даних, яка є основою для створення узагальненого портрету шахрая з метою спрощення процесів їх виявлення.

3. Вперше розроблено узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків, відмінність якого полягає у використанні запропонованої моделі класифікації користувачів та методу подолання різнорідності вхідних даних, що дозволяє визначити класи користувачів та підвищити точність виявлення шахрайства при інсталюванні мобільних додатків.

Практичне значення отриманих результатів роботи полягає у наступному:

- здійснено класифікацію різнорідних даних, що дозволило спростити процес аналізу різнорідних за метриками, розмірностями і шаблонами даних та автоматизувати його;
- розроблено алгоритм пошуку аномалій в даних, алгоритми процесу подолання різнорідності вхідних даних, алгоритм виявлення шахрая при інсталюванні мобільних додатків, алгоритм створення узагальненого портрету шахрая та алгоритм мінімізації часу виявлення шахраїв на основі розпаралелення обчислювальних процесів, які покладені в основу інформаційної технології, що підвищило точність та швидкодію виявлення шахраїв;
- запропоновано інформаційну технологію виявлення шахрайства при інсталюванні мобільних додатків, яка використовує запропоновані метод виявлення шахрайства при інсталюванні мобільних додатків, метод подолання різнорідності вхідних даних, модель класифікації даних, і, на відміну від існуючих систем Antifraud, дозволила підвищити точність

класифікації користувачів до 99,14 %, зокрема точність класифікації шахраїв – до 82,76 %;

– розроблено програмне забезпечення “Mobile App Install Fraud Detection System” для виявлення шахрайства при інсталюванні мобільних додатків.

Результати дисертаційної роботи впроваджені на підприємствах та навчальних закладах: Garuda AI B. V. (м. Схіпгол, Нідерланди) – інформаційна технологія; ТОВ «ВІН ІНТЕРАКТИВ» (м. Вінниця, Україна) – алгоритми подолання різнорідності даних, модель процесу подолання різнорідності даних; ТОВ «4ХайТек» (м. Вінниця, Україна) – модель процесу подолання різнорідності вхідних даних, методика виявлення шахрая при інсталюванні мобільних додатків; ПП «Літсофт» (м. Київ, Україна) – алгоритм подолання різнорідності даних, узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків, методика створення узагальненого портрету шахрая; у навчальний процес кафедри комп’ютерних наук Вінницького національного технічного університету (ВНТУ) – узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків та у навчальний процес кафедри інформатики, програмної інженерії та економічної кібернетики Херсонського державного університету – інформаційна технологія виявлення шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних.

Особистий внесок здобувача. Усі результати, які складають основний зміст дисертації, отримані здобувачем самостійно. У роботах [16], [18], [23], [24], [27], [62] здобувачеві належать усі теоретичні та практичні результати. У роботах, опублікованих у співавторстві, здобувачу належать: [14] – розроблено метод виявлення шахрайства, математичну модель процесу шкалювання, алгоритм шкалювання різнорідних масивів, алгоритм обробки отриманих груп однорідних даних, схему процесу виявлення шахраїв, інтелектуальну систему автоматичного виявлення шахраїв, здійснено класифікацію різнорідних даних при інсталюванні мобільних додатків; [108] – здійснено формалізацію процесу виявлення шахрайства як аномалії в даних, розроблено метод виявлення

аномалій при інсталюванні мобільних додатків; [17], [26] – розроблено метод формування портрету шахрая та алгоритм розробки нечіткої моделі для формування портрету шахрая; [28] – запропоновано інтелектуальну інформаційну технологію виявлення шахрайства при інсталюванні мобільних додатків, удосконалено класифікацію користувачів з використанням глибинних нейронних мереж, створення узагальненого портрету шахрая; [19] – розроблено систему виявлення шахрайства при інсталюванні програмних додатків; [13] – запропоновано метод, моделі та алгоритми подолання різнорідності даних для виявлення шахрайства; [15] – розроблено метод та алгоритм аналізу різнорідних даних, математичну модель процесу аналізу різнорідних даних, схему експериментального дослідження виявлення аномалій в різнорідних даних; [92], [93] – розробка програмного забезпечення для модулю збору даних та для модулю визначення схожості користувачів, розробка алгоритму мінімізації часу виявлення шахраїв, алгоритму пошуку аномалій в даних; [20] – запропоновано модель класифікації користувачів; [22] – розроблено метод виявлення шахрайства при інсталюванні мобільних; [25] – запропоновано метод подолання різнорідності вхідних даних; [21] – запропоновано модель визначення шахрайських способів встановлення мобільних додатків.

Апробація матеріалів дисертації. Результати дисертаційної роботи доповідались та обговорювались на 9 науково-технічних конференціях: XLV, XLVI, XLVIII науково-технічних конференціях професорсько-викладацького складу, співробітників та студентів Вінницького національного технічного університету; науково-практичній конференції «Сучасні тенденції розвитку системного програмування» (м. Київ, Національний авіаційний університет, 2016 р.); XIV Міжнародній конференції «Контроль і управління в складних системах (КУСС-2018)» (м. Вінниця, Вінницький національний технічний університет, 2018 р.); V Міжнародній науково-технічній конференції студентів, магістрів та аспірантів «Інформатика, управління та штучний інтелект» (м. Харків, Національний технічний університет «Харківський політехнічний інститут», 2018 р.); 5th International Winter School on Big Data BigDat2019 (м.

Кембридж, University of Cambridge, Великобританія, 2019 р.); 577th International Conference on Innovative Engineering Technologies (ICIET) (м. Бангкок, Таїланд, 2019 р.); The 10th International Conference on Dependable Systems, Services and Technologies (DESSERT'2019) (м. Лідс, Великобританія, Leeds Beckett University, 2019 р.), де отримано нагороду «Best Paper Award»; The 14th International conference "Computer sciences and Information technologies" (CSIT 2019) (м. Львів, Україна, вересень 17-20, 2019); The 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), (м. Метц, Франція, вересень 18-21, 2019).

Публікації. За темою дисертації опубліковано 20 праць, в тому числі 5 статей надруковано у наукових виданнях, які входять до переліку фахових видань з технічних наук, затверджених МОН України (одна з яких також входить до наукометричної бази даних Scopus). Крім того, 6 статей опубліковано в міжнародних наукових виданнях, п'ять з яких входять до міжнародної наукометричної бази Scopus (три з яких також входять до міжнародної наукометричної бази IEEE Xplore), 7 робіт опубліковано у збірках матеріалів конференцій (три з яких міжнародні), отримано 2 свідоцтва про реєстрацію авторського права на твір.

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел і додатків. Основний зміст викладено на 163 сторінках друкованого тексту, містить 63 рисунки, 17 таблиць. Список використаних джерел містить 108 найменувань. Загальний обсяг 245 сторінки.

РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ ТА ПОСТАНОВКА ЗАДАЧІ ВИЯВЛЕННЯ ШАХРАЙСТВА ПРИ ІНСТАЛЮВАННІ МОБІЛЬНИХ ДОДАТКІВ

1.1 Аналіз об’єкту дослідження

Для дослідження, моделювання та ефективного вирішення задачі виявлення шахрайства при інсталюванні мобільних додатків необхідно здійснити аналіз об’єкту дослідження, визначити поняття шахрайства при інсталюванні мобільних додатків та здійснити аналіз характеристик та властивостей шахрайства як аномалій в даних.

1.1.1 Визначення поняття шахрайства при інсталюванні мобільних додатків

Для сучасного IT-ринку характерно просування компаніями-розробниками мобільних додатків. На таку процедуру компанія повинна витрачати достатньо великі кошти на маркетингові кампанії. Один із варіантів визначення дієвості маркетингової кампанії полягає у перевірці приведеної нею кількості інсталювань мобільного додатку. Саме на цьому кроці варто знати, що частина або вся множина інсталювань мобільних додатків могла бути здійснена шахрайським способом. Знаючи дійсну кількість органічних інсталювань мобільного додатку та кількість інсталювань, здійснених шахрайським способом, можна визначити реальну вартість маркетингової кампанії та її ефективність. Зазначимо, що користувачів, які займаються шахрайством, називають шахраями. Тому актуальною в цьому напрямку є задача розробки інформаційної технології для автоматичного виявлення шахраїв та, відповідно, маркетингових кампаній, які використовують шахрайські способи інсталювання.

Тому розглянемо поняття шахрайства в області інсталювання мобільних додатків. Як зазначено у [30], шахрайство при інсталюванні мобільних додатків,

в основному, передбачає встановлення підозрілих програм, де інсталятори не мають наміру фактично використовувати додаток. Такий вид шахрайства зосереджується на недобросовісних партнерах або філіях, які отримують дохід за приведення інсталювань, незважаючи на те, що вони не прийняли жодної участі у впровадженні програми та приведенні інсталювань, здійснених органічними (справжніми) користувачами.

Можна виділити такі шахрайські способи під час інсталювання мобільних додатків:

1. Кліковий спам (Click Spamming) [1]-[3]. Цей спосіб відноситься до програм, які генерують підроблені запити кліків програмним способом. Тут шахрай спочатку запускає тести, щоб отримати URL-маркери (URL tokens) відповідної маркетингової кампанії. Після цього на кожній ітерації виконуються програми для генерації великого обсягу запитів клікання, рандомізації ідентифікаторів пристроїв (device-IDs) і програмних агентів (user agents). Основна задача полягає в тому, щоб заробити кошти від цих запитів і, на додаток, випадковим чином згенерувати потенційно-органічні установки. Типові наслідки цього сценарію: наявність величезних обсягів кліків, низький CVRs (conversion rate – відношення кількості відвідувачів сайту, які виконали на ньому певні цільові дії до загального числа відвідувачів сайту у відсотках), велика кількість інсталювань програмних додатків і великий трафік, який надійшов через декілька точок мережі чи сайтів. Варто зауважити, що click-спаму в інтернеті, де неетичні видавці використовують шкідливі або обманні шляхи, зараз дуже багато.

2. Мобільне викрадення (Mobile Hijacking) [1] відбувається, коли справжній мобільний додаток для користувача виконує деякі несанкціоновані дії, наприклад, запускає приховані оголошення і формує кліки від імені користувача у фоновому режимі. У цілому, програма буде працювати за сценарієм, що максимально імітує поведінку людини. Проблема полягає в тому, що усі загальні точки даних (IP-адреса і програмний агент, а також ідентифікатор пристрою) здаються дійсними. Основна задача полягає в тому, що потенційно

органічні інсталювання від цього користувача будуть неправильно прив'язані до компанії-розробника мобільного додатку. Цей сценарій буде генерувати велику кількість інсталювань мобільних додатків через кліки і покази та характеризуватиметься низькими CVRs і кращою за середню активністю користувачів. Крім того, з огляду на те, що шахрай не може контролювати, коли установка органічно проводиться користувачем, аналіз проміжку часу від кліка до установки призведе до дуже атипових розподілів на рівні користувача (видавця – publisher або IP-адреси). На рис. 1.1 [2] представлено приклад виконання мобільного викрадення як одного із способів шахрайства при інсталюванні мобільних додатків.

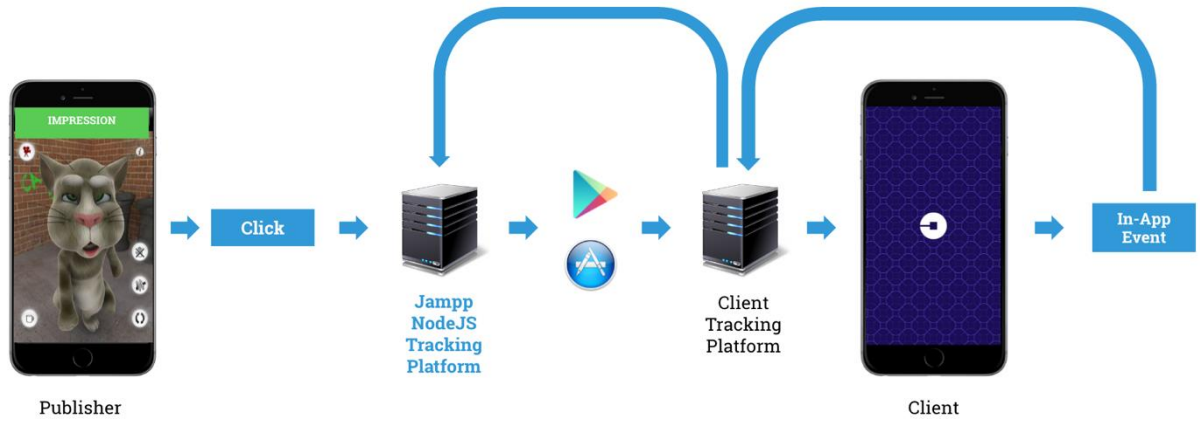


Рисунок 1.1 – Приклад виконання мобільного викрадення як одного із способів шахрайства при інсталюванні мобільних додатків

3. Ферми дій (Action Farms) [3]. Шахраї винагороджують людей по всьому світу за встановлення мобільних додатків у ручному режимі, тобто фактично наймають людей для того, щоб вони встановлювали мобільні додатки. Кількість подій, яка буде досягнута за цією схемою, безумовно, не буде настільки ж великою, як у програмних схемах, але у фінансовому аспекті ця схема набагато дорожча, ніж попередні.

Серед сучасних та найбільш очевидних контрзаходів до шахрайських способів інсталювання мобільних додатків, які вже стали своєрідним стандартом в усій галузі, можна виділити: фільтрацію IP-адреси, блокування видавця,

зовнішню фільтрацію кліків, визначення частоти кліків або запитів інсталювання, визначення співвідношення населення по геолокації (шляхом порівняння дійсної кількості населення у вказаній геолокації з кількістю інсталювань з цієї геолокації), аналіз проміжку часу між подіями (використовують, наприклад, такі відомі фірми як Adjust та Kochava [3], [19], [20]). Тобто усі вказані способи шукають певні відхилення у вхідних даних.

1.1.2 Визначення задачі виявлення шахрайства як однієї із задач пошуку аномалій в даних

Як показано в пункті 1.1.1, шахрайство – це практично аномалія (викиди) в даних. В таких роботах як [30] виявлення шахрайства теж розглядається як одна із підзадач виявлення аномалій. Тому і в даній роботі будемо розглядати задачу виявлення шахрайства як задачу пошуку аномалій в даних.

Розглянемо визначення поняття аномалії різними науковими школами. Так, вчені з університету Мінессота визначають у [30], що аномалія – це шаблон даних, який не відповідає визначеному поняттю нормальної поведінки (заданому шаблону). При цьому аномалії можуть бути наявними у вхідних наборах даних систем прийняття рішень та систем штучного інтелекту через неуважність персоналу, який вносив ці дані, через наявність похибок та некоректну роботу системи збору даних або ж із-за навмисних дій шахраїв (іншими словами – шахрайство).

Розглянемо з технічної точки зору, що в даній роботі означає «аномалія» та «шахрайство». Поняття аномалії згадується в літературі з Data Mining (аналізу даних) [31]-[34] як нетипова поведінка, аномальність, викид (outliers), відхилення. Про це також зазначають вчені з Тернопільського національного економічного університету. Існують визначення, які вважаються досить загальними. Так наприклад Хокінз (Hawkins, 1980) у [35] визначає аномалію як спостереження, яке відхиляється від інших спостережень настільки, що виникають підозри, що воно було породжено іншим механізмом. Барнет і Льюїс

(Barnet and Lewis, 1994) вказують на те, що аномалія є тим, що помітно відхиляється від інших зразків, в яких воно виявляється. Аналогічно, Джонсон (Johnson, 1992) визначає аномалію як спостереження у наборі даних, яке суперечить іншій частині цього набору даних. Але в даних прикладах не дається чітке визначення поняття аномальності, яке може бути використане для розробки математичних моделей та програмного забезпечення. З іншого боку, простий приклад аномалії у двовимірному просторі представлено на рис. 1.2 [30]. На рис. 1.2 є дві області з нормальними даними, оскільки більшість спостережень лежить у цих двох регіонах. Точки, які знаходяться досить далеко від цих двох регіонів, наприклад, точки O_1 та O_2 та точки в регіоні O_3, є аномаліями.

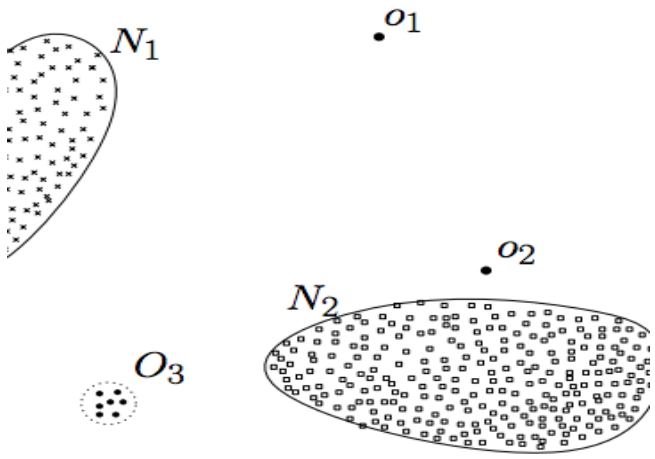


Рисунок 1.2 – Простий приклад аномалії у двовимірному просторі

У даній роботі будемо вважати, що аномалія в даних – це набори групи даних, які належать заданій вибірці, але значення яких відрізняються від заданих характеристик.

Шахрайство ж у даній роботі визначатимемо як навмисне породження аномалії в даних сторонньою особою (шахраєм) або механізмом з певною метою. Аномалії, як і шахрайство, можуть бути у всіх наборах даних, а отже й у всіх сферах, з якими працюють системи прийняття рішень та системи штучного інтелекту, а також у наборах даних про користувачів при інсталюванні

мобільних додатків. На рисунку 1.3 показано деякі із сфер, в яких може мати місце шахрайство, саме як навмисне введення аномалій у набори даних.

Рисунок 1.3 – Приклад предметних областей, в яких має місце шахрайство

Шахрайські та аномальні дані можуть бути наявні і в наступних областях:

– виявлення несправностей у механізмах;

– виявлення вторгнень (Intrusion Detection);

– виявлення нестандартних гравців на біржі (інсайдерів), тощо.

Також на рис. 1.4 – 1.5 показано характеристики, за якими може бути виявлено шахрайство у деяких з представлених областей.

На основі інформації, що представлена на рис. 1.3 – 1.5, для кожної з областей, у вхідних наборах даних яких має місце шахрайство, можна виділити вимоги. Так, наприклад, виділимо вимоги до виявлення шахрайських (аномальних) даних у банківських транзакціях (рис. 1.4):

– геолокація транзакції повинна бути звичною для користувача, інакше це можна задати правилом $F_1(current\_location) = (current\_location \in \{loc_1, loc_2, ..., loc_N\})$;

– якщо користувач робить нетипову для себе покупку, то ця транзакція може бути шахрайською. Тому необхідно задати правило $F_2(current\_type)$, яке перевірятиме, чи тип проведеної транзакції є звичним для користувача, тобто $current\_type \in \{type_1, ..., type_N, ...\}$;

– час між транзакціями T_{avg} повинен бути звичним для органічних користувачів, які виконують органічні (не шахрайські) транзакції. Дану вимогу можна задати правилом $F_3(T_{avg})$, яке визначає, що T_{avg} повинно бути у проміжку між мінімальним $T_{avg}\min$ та максимальним $T_{avg}\max$ значеннями часу між органічними транзакціями.

Отже, виявлення шахрайства у банківських транзакціях можна задати наступним правилом:

$$F_1(current\_location) \| F_2(current\_type \| F_3(T_{avg}).$$

Оскільки шахраї створюють нові способи інсталювання мобільних додатків та шахрайства у інших розглянутих областях для того, щоб обійти існуючі системи виявлення аномалій та шахрайства, то доцільно було б створити інформаційну технологію, яка класифікує дії та користувачів не лише за заданими правилами, а базуючись на всіх вхідних даних. Проте з наведених характеристик шахрайських та неаномальних даних видно, що усі шахрайські дані мають спільні ознаки, оскільки неможливо придумати мільярди різних способів шахрайства, тому, зважаючи на це, можна зробити припущення, що всі неаномальні дані можна буде класифікувати в один клас, а аномальні – в інший, базуючись на їх спільних ознаках.

Перед тим як розглянути методи вирішення задач виявлення аномалій у наборах даних, області застосування яких зображено на рис. 1.3, розглянемо

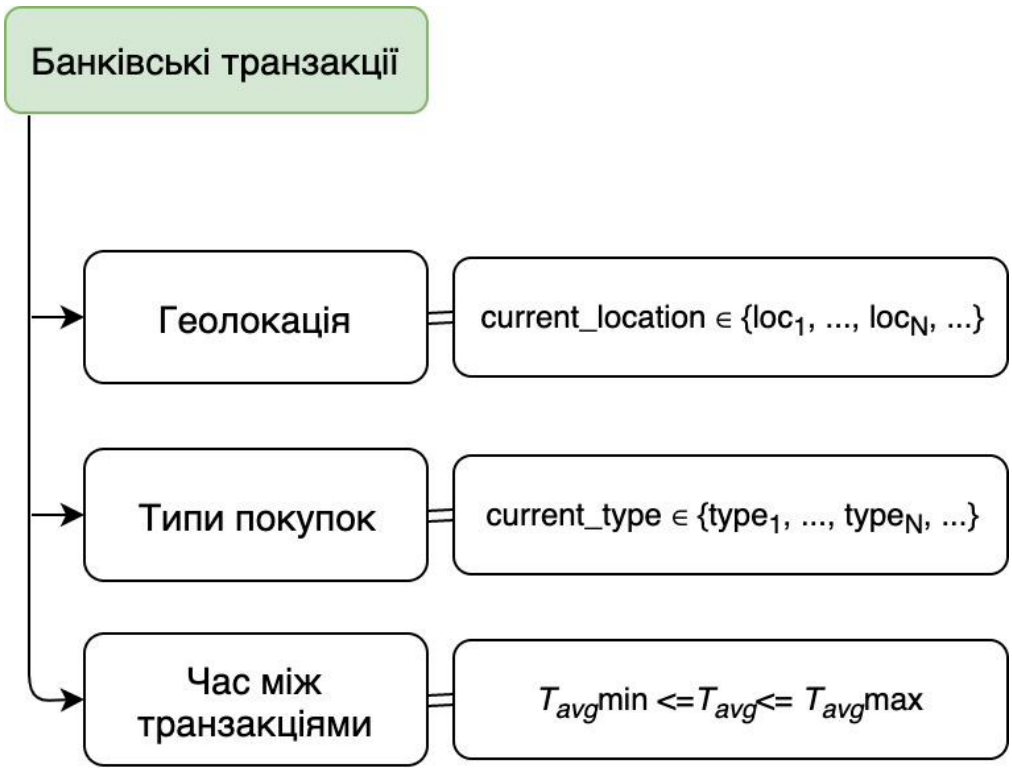


Рисунок 1.4 – Характеристики, за якими може бути виявлено шахрайство у банківських транзакціях

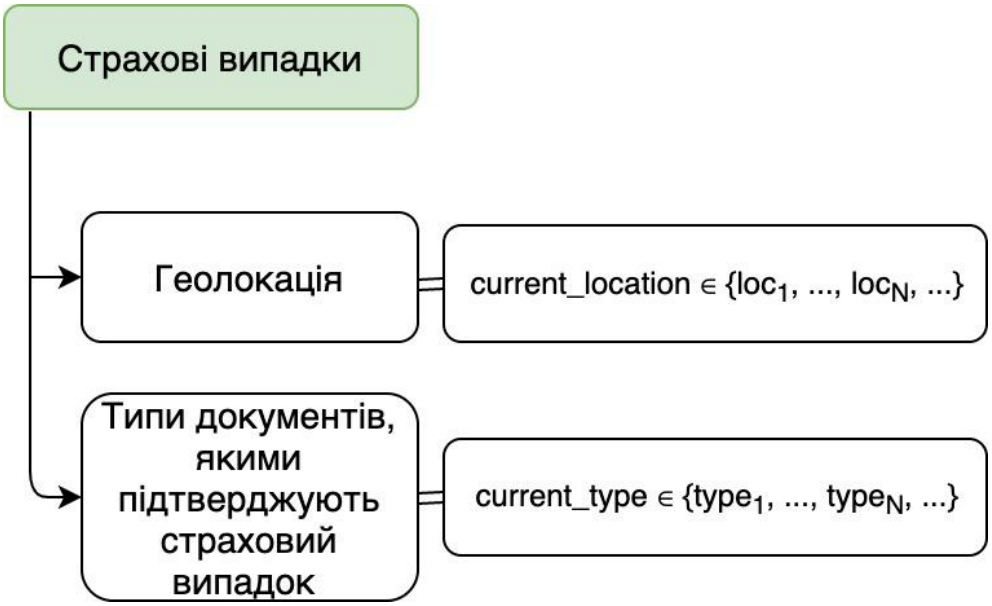


Рисунок 1.5 – Характеристики, за якими може бути виявлено шахрайство у страхових випадках

критерії класифікації шахрайства як аномалії в даних, а саме – проаналізуємо його характеристики та властивості. Правильно прокласифікувавши шахрайство як аномалію в даних, можна вибрати найефективніший метод саме для поставленої задачі (предметної області).

1.1.3 Аналіз характеристик та властивостей шахрайства як аномалії в даних

Оскільки задача виявлення шахрайства відноситься до класу задач пошуку аномалій, як зазначено у пункті 1.1.2, доцільно проаналізувати критерії класифікації аномалій, а саме – їх характеристики та властивості.

Аналіз характеристик та властивостей шахрайства як аномалії в даних є необхідним, тому що проблему виявлення аномалії в її найбільш загальній формі нелегко вирішити, а такий аналіз допомагає вибрати найбільш доцільний метод вирішення задачі. Фактично, більшість існуючих методів виявлення аномалії вирішують конкретне формулювання проблеми. Формулювання спричинене різними факторами, такими як характер даних, наявність помічених даних, тип аномалій, які слід виявляти тощо. Часто ці фактори визначаються областю застосування, в якій потрібно виявляти аномалії. Дослідники запозичили концепції з різних дисциплін, таких як статистика, машинне навчання, інтелектуальний аналіз даних, теорія інформації, спектральна теорія, і застосували їх до конкретних формулювань проблем. На рис. 1.6 показано вищезазначені основні компоненти, пов'язані з будь-якою технікою виявлення аномалій в даних, а також – виявлення шахрайства [30].

Розглянемо критерії (властивості та характеристики), за якими можна класифікувати шахрайство як аномалію в даних (доцільно зауважити, що ці критерії слід врахувати у процесі розробки інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків):

1. Характер вхідних даних (Nature of Data). Ключовим аспектом будь-якого методу виявлення аномалій є характер (природа) вхідних даних. Вхід, як

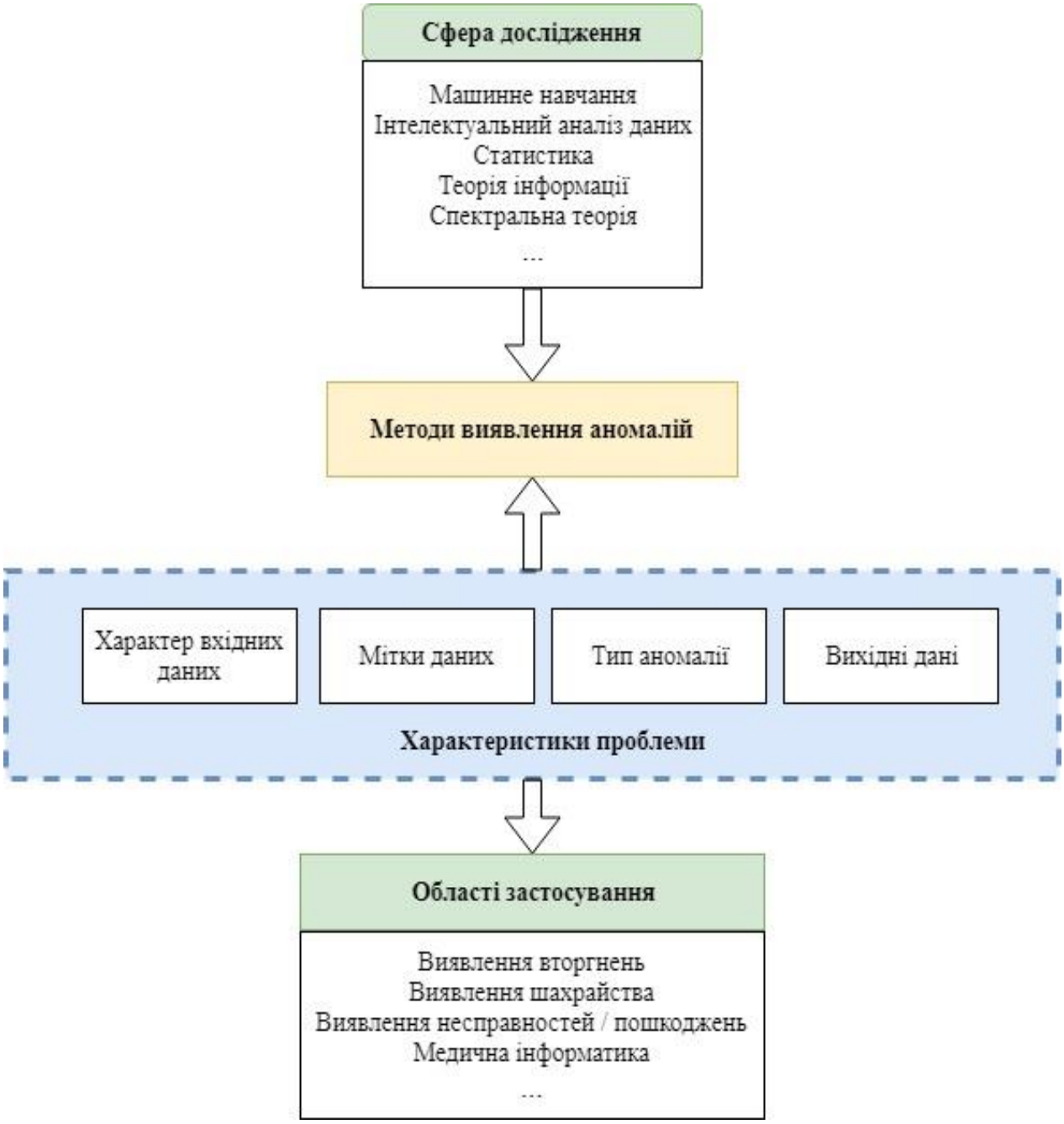


Рисунок 1.6 – Ключові компоненти, пов’язані з методами виявлення аномалій

Атрибути можуть бути різних типів, такі як бінарні, категорійні або безперервні. Кожен екземпляр даних може складатися тільки з одного (одновимірний) або кількох (багатовимірний) атрибутів. У разі багатовимірних екземплярів даних, усі атрибути можуть бути або одного типу, або сумішшю різних типів даних. Характер (природа) атрибутів визначає придатність методів виявлення аномалій, зокрема шахрайства, у конкретній задачі [30].

2. Тип аномалії (Anomaly Type). Важливим аспектом методів виявлення

аномалій є тип аномалії.

Так, в [30] запропонована наступна класифікація аномалій:

– точкові аномалії (point anomalies) мають місце, якщо окремий екземпляр даних можна розглядати як аномальний по відношенню до решти даних;

– контекстні (умовні) аномалії (contextual anomalies) мають місце, якщо екземпляр даних є аномальним у конкретному контексті (але не інакше);

– колективні аномалії (collective anomalies) мають місце, якщо набір пов'язаних примірників даних є аномальним по відношенню до всього набору даних.

3. Мітки даних (Data Labels). Мітки, пов'язані з екземпляром даних, позначають цей екземпляр нормальним або аномальним. Слід зазначити, що отримання таких міток часто обходиться занадто дорого. Виставлення міток часто робиться вручну за допомогою людини-експерта, а отже, вимагає значних зусиль, щоб отримати помічений набір навчальних даних. Як правило, отримання поміченого набору аномальних примірників даних, які охоплюють усі можливі типи аномальної поведінки є більш складним, ніж отримання мітки для нормальної поведінки. Крім того, аномальна поведінка часто має динамічний характер, наприклад, можуть виникнути нові типи аномалій, для яких немає даних, помічених під час навчання. На підставі того, наскільки мітки доступні, методи виявлення аномалій можуть працювати в одному з таких трьох режимів [36]:

– контрольоване виявлення аномалій (виявлення аномалій з учителем – supervised anomaly detection). Методи контрольованого виявлення аномалій припускають наявність великої кількості навчальних даних, які мають екземпляри для нормального класу, а також класу аномалій. У межах розв’язання задачі виявлення шахрайства при інсталюванні мобільних додатків цей режим виявлення аномалій не є ефективним, оскільки шахрайські способи можуть з’являтися кожного дня, і неможливо створити базу з навчальними даними, яка б покривала усі випадки;

– напівконтрольоване виявлення аномалій (semi-supervised anomaly

detection). Методи, які працюють у даному режимі, припускають, що тренувальні дані позначають примірники тільки для нормального – неаномального – класу. Тому вони мають більш широке застосування, ніж контрольовані методи, оскільки не вимагають мітки для примірників класу аномалій;

– виявлення аномалій без нагляду (некотрольовані алгоритми – unsupervised anomaly detection). Методи, які працюють у неконтрольованому режимі, не вимагають підготовки даних, і, таким чином, найбільш широко застосовуються. Методи у цій категорії роблять неявне припущення, що нормальні екземпляри зустрічаються набагато частіше, ніж аномальні екземпляри у тестових даних. Якщо це припущення неправильне, то такі методи характеризуються високим рівнем помилкових тривог. Багато напівконтрольованих методів може бути пристосовано для роботи у неконтрольованому режимі, використовуючи зразок неміченого набору даних у якості навчальних даних. Така адаптація передбачає, що дані випробувань містять дуже мало аномалій і модель, отримана у ході навчання, є стійкою до цих кількох аномалій.

2. Вихідні дані виявлення аномалій (Output of Anomaly Detection). Важливим аспектом для будь-якого методу виявлення аномалій є спосіб повідомлення про аномалії. Як правило, кінцевими сигналами методів виявлення аномалій є один з таких двох типів [30]:

– рейтинг (scores) – методи рейтингування призначають аномальний результат кожному примірнику в тестових даних залежно від того, в якій мірі цей екземпляр вважається аномалією. Отже, виходом у таких методах є ранжований список аномалій. Аналітик може вибрати або проаналізувати кілька верхніх аномалій або використати поріг відсікання для вибору аномалії;

– мітки (labels) – методи у цій категорії прикріплюють мітку (нормальний або аномальний) для кожного екземпляру тесту.

Методи рейтингування для виявлення аномалій дозволяють аналітику використовувати певний поріг, щоб вибрати найбільш значимі аномалії. Методи, які забезпечують бінарні мітки для тестових екземплярів, безпосередньо не

дають змогу аналітикам зробити такий вибір, хоча це можна контролювати додатково через вибір параметрів у рамках кожного методу [30].

Оскільки у задачі виявлення шахрайства при інсталюванні мобільних додатків має місце робота з Big Data, то доцільно обрати метод, який би ранжував вихідну інформацію.

З вищенаведеного аналізу аномалій, до яких відноситься і шахрайство, можна зробити висновок, що для розроблення системи виявлення шахрайства при інсталюванні мобільних додатків слід визначити природу існуючих та можливих шахрайських способів інсталювання додатків.

1.2 Варіантний аналіз методів пошуку аномалій в даних

Як було зазначено у підрозділі 1.1, задача виявлення шахрайства у даній роботі розглядається як задача пошуку аномалій в даних. Тому здійснимо варіантний аналіз існуючих методів пошуку аномалій в даних.

Аналізуючи методи, слід зазначити, що як згадувалося раніше у пункті 1.1.3, конкретне формулювання проблеми визначається кількома різними критеріями, такими як характер вхідних даних, доступність (або недоступність) міток, а також обмеження та вимоги, властиві для сфери застосування [30]. Вказані критерії показують множинність проблемної області і виправдовують необхідність широкого спектру методів виявлення аномалій.

1.2.1 Класифікація методів виявлення аномалій

Аналіз джерел [30]-[34], [36]-[58] дозволив здійснити класифікацію методів, що використовуються на даний час для виявлення аномалій. Всі методи можна розділити на такі: методи класифікації, методи кластеризації, статистичні методи, мікс методів (рис. 1.7).

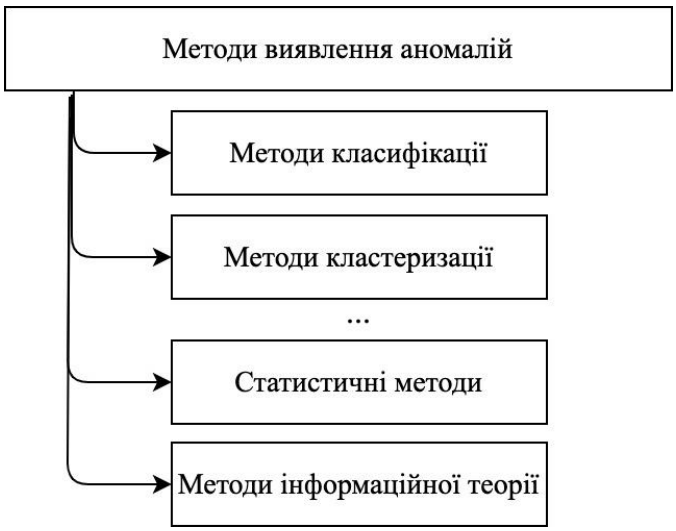


Рисунок 1.7 – Методи виявлення аномалій

Оскільки вибір методу виявлення аномалії залежить від поставленої задачі, розглянемо підзадачі виявлення аномалій (рис. 1.8). Серед них є виявлення кібер-атак, виявлення шахрайства, виявлення медичних аномалій, виявлення промислового пошкодження, оброблення зображення, виявлення текстової аномалії, сенсорні мережі тощо [30].

До методів виявлення аномалій відносяться методи інтелектуального аналізу даних (Data Mining). Основними задачами інтелектуального аналізу даних є [59]:

– класифікація – її завданням є визначити приналежність об'єкта до класу по його характеристикам. Необхідно зауважити, що в цьому завданні існує безліч заздалегідь відомих класів, до яких може бути віднесений об'єкт;

– регресія – пошук функції, яка описує залежність між характеристиками об'єкта з найменшою похибкою;

– пошук асоціативних правил;

– кластеризація – виявлення в даних прихованої структури, яка так чи інакше схожа;

– виявлення нетипових спостережень – виявлення в даних нетипових спостережень, які представляють «особливий» інтерес або виявлення помилок, від яких необхідно позбутися для проведення подальшого аналізу.

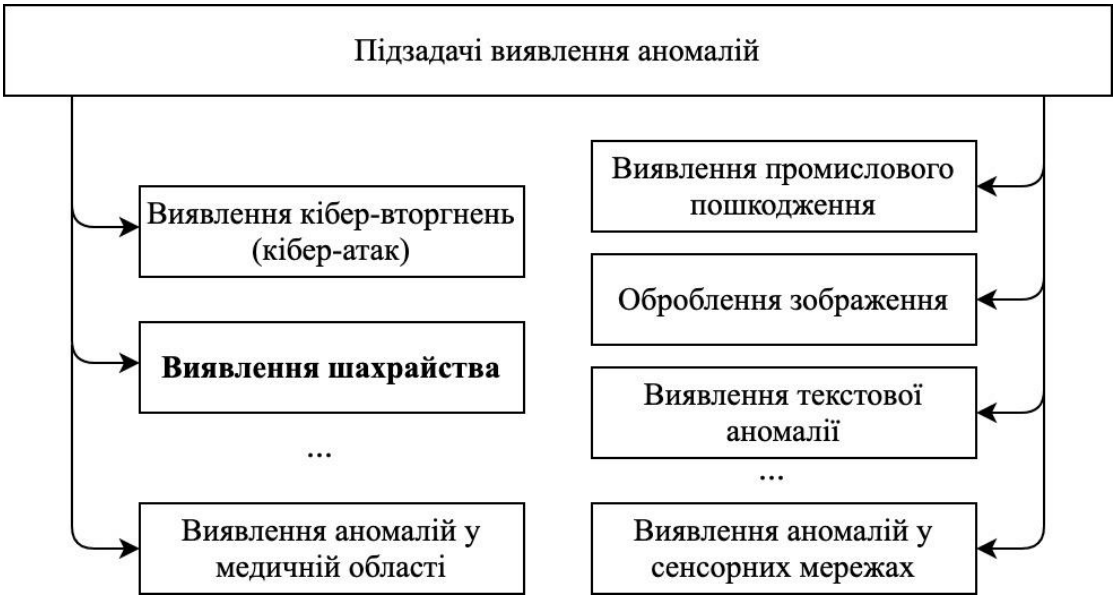


Рисунок 1.8 – Області виявлення аномалій

Загалом, інтелектуальний аналіз даних – це дослідження і виявлення «машиною» (алгоритмами, засобами штучного інтелекту) в сирих даних прихованих знань, які раніше не були відомі, нетривіальні, практично корисні, доступні для інтерпретації людиною. Засновник напрямку Data Mining – Пятецкий-Шапіро [59].

Врахувавши усе вище вказане, у ході дослідження методів виявлення аномалій, було проаналізовано джерела [29]-[54] та виділено методи машинного навчання, які можна поділити на:

– методи класифікації, багато з яких згадується, наприклад, у роботі [31]. Також тут можна виділити експертні системи, які використовуються при виявленні аномалій в медичній області, в кредитних картках, при обробці зображень, при виявленні вторгнень у мережу. Такі системи розглядаються, наприклад, в роботі [40]. Проте недоліком експертних систем є те, що при появі нових шахрайських шаблонів необхідно самостійно відслідковувати такі шаблони та дописувати нові правила у систему. Окремо виділяють баєсову мережу, яка використовується при виявленні аномалій у медичних даних, обробці зображень, сенсорних мережах. Баєсова мережа згадується, наприклад,

у роботах [32], [41]-[43]. Також існують такі методи як Support Vector Machine, метод k-найближчих сусідів, методи класифікації на основі нейронних мереж. Останній метод найчастіше використовується при виявленні шахрайства у кредитних картках, при обробці зображень, при виявленні вторгнень у мережу, згідно до роботи [30]. Але як показано у розглянутих джерелах ці техніки ефективні при використанні однорідних даних;

– методи кластеризації, які поділяють на ієрархічні (таксономія) та неієрархічні або ж на чіткі та нечіткі. Серед методів даної групи виділимо k-means clustering, графові методи, серед яких є, наприклад, алгоритм виділення зв'язних компонент. Також можна виділити алгоритм ФОРЭЛ та алгомеративну ієрархічну кластеризацію. Більшість вказаних методів розглянуто у роботі [33] та у професійному інформаційно-аналітичному ресурсі [44]. Також необхідно зазначити, що один із способів кластеризації базується на використанні коефіцієнтів схожості і це показано, наприклад, у роботах авторів [19]-[22], [63] та у роботі [45]. Методи кластеризації найчастіше використовуються при network intrusion detection, що розглядається в роботі [46], та при credit card fraud detection, але особливістю таких методів є використання лише однорідних даних;

– статистичні методи, які розглядаються, наприклад, у описах (surveys) [47], [48]. У даній групі методів виділяють спектральний метод, який найчастіше використовується при Mobile Phone Anomaly Detection та виявленні аномалій у сенсорних мережах. Також виділяють непараметричне статистичне моделювання, яке використовується при виявленні вторгнень у мережу, виявленні неполадок у механізмах. Ще один метод, який відноситься до даної групи, – параметричне статистичне моделювання, яке розглядається в роботах [49], [50]. Статистичне профілювання з використанням гістограм, що розглядається в роботі [51], також відноситься до даної групи методів. Проте при використанні даних методів відсутні процедури зведення даних до однорідних.

Зокрема, методи кластеризації не підходять для вирішення задачі виявлення шахрайства при інсталюванні мобільних додатків, оскільки це є методи навчання без учителя, тобто методи, які самі виявляють групи схожих

користувачів. Проте у задачі, що розглядається, необхідно наперед визначити класи, по яким повинно проводитись групування користувачів.

В результаті аналізу існуючих методів виявлення аномалій та областей їх застосування, в роботі була розроблена таблиця 1.1. Більшість методів, що розглянуті в підрозділі 1.2, представлено у даній таблиці.

Серед усіх можливих методів виявлення аномалій розглянемо ті, які найбільше підходять до вирішення поставленої задачі виявлення аномальних та шахрайських даних у вхідних даних систем прийняття рішень та систем штучного інтелекту. Для цього розглянемо різні сфери, у яких мають місце аномалії у вхідних наборах даних, та в яких наявність аномалій може призвезти до серйозних збитків та втрат. Так, наприклад,

– у ході досліджень фізичних процесів використовуються статистичні методи, експертні системи, нейронні мережі тощо. Прикладом аномалій у даній сфері може бути, наприклад, випадок, коли у вхідному наборі даних визначення опору навмисно змішані дані змінного струму та напруги.

– у медичних процесах – нейронні мережі, експертні системи, параметричне статистичне моделювання, баєсова мережа, методи на основі методу найближчих сусідів;

– у радіотехнічних системах, системах управління технологічними процесами, у мережах і системах телекомунікації аномалією можна вважати білий шум (викиди), які частіше всього не враховують та не аналізують їх природу, характер і причину появи. У вказаних системах для пошуку аномалій ефективно використовувати нейронні мережі, експертні системи тощо.

Для вибору найбільш ефективних методів виявлення шахрайства розглянемо наступні аспекти [30]:

– характер вхідних даних,

– труднощі, пов’язані із виявленням аномалій у поставленій задачі,

Таблиця 1.1

Класифікація методів виявлення аномалій за сферами їх найчастішого використання

Методи/область вхідних наборів даних	Медичні дані	Виявлення інсайдерської торгівлі	Кредитні картки	Обробка зображень	Інсталювання мобільних додатків	Виявлення вторгнень на основі хосту	Виявлення вторгнень у мережу	Мобільні телефони	Виявлення неполадок у механізмах	Виявлення структурних пошкоджень	Текстові дані	Сенсорні мережі
Нейронні мережі	+		+	+		+	+	+	+	+	+	
Статистичне профілювання з використунням гістограм		+				+	+	+		+	+	
Мікс моделей				+		+				+	+	
Метод опорних векторів				+		+	+				+	
Експертні системи (на основі правил)	+		+		+	+	+	+	+			+
Кластеризація			+	+			+				+	
Параметричне статистичне моделювання	+						+	+	+	+		+
Непараметричне статистичне моделювання							+		+			
Баєсова мережа	+			+			+					+
На основі методу найближчих сусідів	+			+			+					+
Методи інформаційної теорії		+										
Спектральні методи									+			+
Регресія				+								

– існуючі методи виявлення аномалій у поставленій задачі.

Фірма StatSoft, яка займається сучасними технологіями аналізу даних, пропонує використовувати метод виявлення нетипових спостережень для виявлення шахрайства [59].

Фосетт та Провост ввели термін моніторинг діяльності (activity monitoring) як загальний підхід до виявлення шахрайства. Типовим підходом методів виявлення аномалій є підтримка профілю використання для кожного клієнта та моніторинг профілів для виявлення будь-яких відхилень. Для аналізу існуючих методів, розглянемо найпоширеніші підзадачі виявлення шахрайства, а саме: виявлення шахрайства з кредитними картками (Credit Card Fraud Detection), виявлення шахрайства з мобільними телефонами (Mobile Phone Fraud Detection), виявлення шахрайства при страхуванні (Insurance Claim Fraud Detection), виявлення внутрішньої торгівлі (Insider Trading Detection). Кожна з цих підзадач різна, але їх розгляд дозволить розширити кругозір при виборі методу виявлення шахрайства при інсталюванні мобільних додатків. Також розглянемо існуючі рішення поставленої задачі [30]:

– виявлення шахрайства з кредитними картками. У цій сфері застосовуються методи виявлення аномалій для виявлення шахрайства з кредитними картками або шахрайського використання кредитних карток (пов'язаних з крадіжками кредитних карток). Виявлення шахрайства з кредитними картками схоже на виявлення шахрайства при страхуванні. Для вирішення цієї задачі використовуються нейронні мережі, кластеризація експертних систем (Rule-based Systems Clustering);

– виявлення шахрайства з мобільними телефонами. Виявлення шахрайства на мобільних/стільникових пристроях – це типова проблема моніторингу діяльності. Завдання полягає у скануванні великого набору облікових записів, вивченні поведінки викликів з/на кожного та видачі сигналу, коли виявлено, що є зловживання обліковим записом. Одним з методів, які вирішують цю задачу, є статистичне профілювання з використанням гістограм, параметричне

статистичне моделювання, нейронні мережі, (експертні системи) на основі правил;

– виявлення шахрайства при страхуванні. Важливою проблемою в галузі індустрії страхування майна є випадки шахрайства, наприклад шахрайство при автомобільному страхуванні. Певні особи маніпулюють системою обробки претензій для несанкціонованих та незаконних позовів. Виявлення такого шахрайства було дуже важливим для страхових компаній, щоб уникнути фінансових втрат. Для вирішення цієї задачі використовують нейронні мережі;

– виявлення внутрішньої торгівлі. Для вирішення цієї задачі використовують статистичне профілювання з використанням гістограм.

Розглянемо підгрупи, на які поділяють методи для використання в задачах виявлення аномалій: методи класифікації, методи кластеризації, статистичні методи, мікс методів на прикладі глибинного навчання.

1.2.2 Методи кластеризації

Розглянемо найрозповсюдженіші методи виявлення аномалій на основі кластеризації.

Обчислювальна складність методів виявлення аномалій на основі кластеризації залежить від алгоритму кластеризації, який використовується для генерації кластерів даних. Таким чином, такі методи можуть мати квадратичну складність, якщо метод кластеризації потребує обчислення попарних відстаней для всіх примірників даних, або лінійну – при використанні евристичних методів, таких як K-means [59], [71] або методів приблизної кластеризації [72]. Фаза тестування методів на основі кластеризації дуже швидка, так як вона включає в себе порівняння тестового примірника з невеликим числом кластерів.

Розглянемо основні переваги та недоліки методів на основі кластеризації [30], [31], [34], [36] у таблиці 1.2.

Таблиця 1.2

Переваги та недоліки методів на основі кластеризації

Методи на основі кластеризації	
Переваги	**Недоліки**
– методи на основі кластеризації можуть працювати в неконтрольованому режимі; – такі методи часто можуть бути адаптовані до інших складних типів даних простим підключенням в алгоритм кластеризації конкретного типу даних, який треба обробляти; – фаза тестування методів на основі кластеризації дуже швидка, так як число кластерів, з якими необхідно порівнювати кожен екземпляр тесту є малою константою.	– продуктивність методів, які базуються на кластеризації, в значній мірі залежить від ефективності алгоритму кластеризації, що використовується для об'єднання нормальних екземплярів у кластерну структуру; – багато методів виявляють аномалії в якості побічного продукту кластеризації, і, отже, не оптимізовані для виявлення аномалій; – деякі алгоритми кластеризації кожен екземпляр примусово призначають до якоїсь групи. Це може призвести до того, що аномалії буде призначено до великого кластеру, таким чином, їх буде розглянуто як звичайні випадки за допомогою методів, які припускають, що аномалії не належать ні до одного кластеру; – кілька методів основаних на кластеризації є ефективними тільки тоді, коли аномалії не утворюють значних скупчень між собою; – обчислювальна складність для кластеризації даних часто є вузьким місцем, особливо якщо використовуються алгоритми $O(N^2d)$ кластеризації.

Зазначимо, що кластеризація – це класифікація без учителя моделей (спостережень, елементів даних або векторів функцій) у групи (кластери) [73].

Важливо розуміти різницю між кластеризацією (класифікацією без учителя) та дискримінантним аналізом (контрольованою класифікацією). У класифікації з учителем на вхід подається колекція маркованих (попередньо класифікованих) даних; проблема полягає в позначенні нового, ще не позначеного шаблону. Як правило, подані позначені (тренувальні) шаблони використовуються для вивчення описів класів, які, у свою чергу, використовуються для позначення нового шаблону. У випадку кластеризації

проблема полягає в тому, щоб згрупувати певну колекцію немаркованих шаблонів у важливі кластери. У певному сенсі, мітки (labels) також пов'язані з кластерами, але ці мітки категорій (category labels) є даними; тобто вони отримуються виключно з даних [30].

Типовий шаблон кластеризації включає в себе наступні етапи [74]: (1) представлення шаблону (необов'язково, включаючи вилучення та/або вибір функції); (2) визначення межі близькості шаблону, відповідного до області даних; (3) кластеризація або групування, (4) абстракція даних (при необхідності), і (5) оцінка результату (при необхідності).

Крок групування можна виконати кількома способами. Вихідні дані кластеризації (або кластеризацій) можуть бути чіткими (hard) (розділ даних на групи) або нечіткими (де кожен шаблон має різну ступінь членства в кожному з вихідних кластерів). Ієрархічні алгоритми кластеризації створюють вкладені рядки розділів, засновані на критерії злиття чи розщеплення кластерів на основі подібності. Алгоритми розбиття визначають розподіл, який оптимізує (як правило, локально) критерій кластеризації. Додаткові методи для групування включають імовірнісні та граф-теоретичні методи кластеризації [30].

Наявність такої великої колекції алгоритмів кластеризації в літературі може змусити користувача спробувати вибрати алгоритм, придатний для поставленої проблеми. В Dubes та Jain [74] для порівняння алгоритмів кластеризації використовується набір критеріїв прийнятності, визначених Фішером та Ван Нессом. Ці критерії прийнятності базуються на: (1) способі формування кластерів; (2) структурі даних; (3) чутливості методу кластеризації до змін, які не впливають на структуру даних. Проте, немає критичного аналізу алгоритмів кластеризації, що стосуються важливих питань, таких як: як дані повинні бути нормалізовані? Яку міру подібності доцільно використовувати в даній ситуації? Як використовувати знання сфери застосування в певній проблемі кластеризації? Як ефективно кластеризувати великий набір даних (скажімо, мільйон моделей)?

Тому потрібно вміти впевнено оцінювати компроміси різних методів та, в кінцевому підсумку, приймати виважене рішення щодо техніки або набору методів роботи в конкретній задачі. Не існує методу кластеризації, загальноприйнятого для розкриття різноманіття структур, присутніх у багатовимірних наборах даних.

1.2.3 Методи класифікації

Розглянемо найпопулярніші та найбільш часто вживані методи виявлення аномалій на основі класифікації [30].

Support Vector Machine (SVM) – використовується для бінарної класифікації. Проте у поставленій задачі може бути набагато більше кластерів, які наперед не можуть бути відомими, оскільки шахрайські способи інсталювання мобільних додатків можуть змінюватись кожного дня. А також, у поставленій задачі необхідно вміти вказувати на причину віднесення до того чи іншого класу.

Обчислювальна складність методів на основі класифікації залежить від використовуваного алгоритму класифікації. Як правило, навчання дерев рішень відбувається швидше, в той час як методи, які пов'язані з квадратичною оптимізацією, такі як SVMs, є більш дорогими, хоча SVMs лінійного часу [64] було запропоновано, щоб мати лінійний час навчання. Фаза тестування методів класифікації, як правило, дуже швидка, так як фаза тестування використовує навчену модель для класифікації.

Розглянемо основні переваги та недоліки даного методу у таблиці 1.3.

Методи на основі пошуку найближчого сусіда [30], [33].

Обчислювальна складність. Основний недолік методу найближчого сусіда і методу LOF – це обчислювальна складність $O(N^2)$, яка виникає через те, що ці методи включають знаходження найближчих сусідів для кожного екземпляра ефективних структур даних, таких як k-d (k-мірних) дерева [66] і R-дерева [67]. Але такі методи не масштабуються, коли кількість атрибутів збільшується.

Проте якщо оцінка аномалії потрібна для кожного екземпляра тесту, такі методи не застосовуються. Методи, які поділяють атрибут простору в гіперсітку, є лінійними за розміром даних, але мають експоненціальну кількість атрибутів, а, отже, не дуже добре підходять для великої кількості атрибутів. Методи відбору проб спробували вирішити проблему зі складністю $O(N^2)$ шляхом визначення найближчих сусідів в межах невеликого зразка набору даних. Але відбір проб може призвести до помилкових оцінок аномалії, якщо розмір вибірки дуже малий. Так, зазначимо, що у випадку, коли $k=1$, даний метод відносить невідомий об'єкт до того класу Ω_l ($l \in \overline{1, M}$), для якого в навчальній вибірці знаходиться «найближчий сусід» ω_i невідомого об'єкта ω:

$$R(\omega, \omega_l) = \min_{i \in \overline{1, N}} R(\omega, \omega_i), \tag{1.1}$$

де $R(\omega, \omega_i)$ – відстань між невідомим об'єктом ω і об'єктом ω_i навчальної вибірки;

N – довжина навчальної вибірки.

Розглянемо основні переваги [68]-[70] та недоліки [30], [68]-[70] даного класу методів у таблиці 1.4.

Таблиця 1.3

Переваги та недоліки методу опорних векторів

Метод опорних векторів	
Переваги	**Недоліки**
– методи на основі класифікації, особливо мультикласові методи, можуть використовувати потужні алгоритми, які здатні відрізняти екземпляри, приналежні до різних класів; – фаза тестування методів на основі класифікації дуже швидка, так як кожен екземпляр тесту порівнюється з попередньо обчисленою моделлю.	– методи на основі мультикласової класифікації використовують наявні точні мітки для різних нормальних класів. Проте часто неможливо отримати дані з такими мітками; – методи на основі класифікації присвоюють мітки кожному тестовому екземпляру, що також може стати недоліком, коли значущий аномальний результат бажаний для тестових екземплярів. Деякі методи класифікації, які отримують оцінку вірогідності прогнозування з виходу класифікатора, можуть бути використаними для вирішення цієї проблеми [67].

Таблиця 1.4

Переваги та недоліки методів на основі пошуку найближчого сусіда

Методи на основі пошуку найближчого сусіда	
Переваги	**Недоліки**
– алгоритм стійкий до аномальних викидів, тому що імовірність включення такого запису в число k-найближчих сусідів мала. Якщо ж це відбулося, то вплив на голосування (особливо зважене) (при k > 2) також, швидше за все, буде незначним, і, отже, малим буде і вплив на підсумок класифікації; – програмна реалізація алгоритму відносно проста; – результат роботи алгоритму легко піддається інтерпретації. Експертам у різних областях цілком зрозуміла логіка роботи алгоритму, заснована на пошуку схожих об'єктів; – можливість модифікації алгоритму, шляхом використання найбільш придатних функцій сполучення і метрик дозволяє підбудувати алгоритм під конкретну задачу.	– 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; – для неконтрольованих методів, якщо дані – це нормальні екземпляри, які не мають достатньо близьких сусідів, або якщо єдині їх сусіди – це аномалії, то в цьому випадку методи не в змозі правильно кластеризувати такі дані; – для напівконтрольованих методів, якщо нормальні екземпляри в тестових даних не мають достатньо схожих нормальних примірників в навчальних даних, відсоток помилкових спрацьовувань таких методів є високим; – обчислювальна складність на етапі тестування також є серйозною проблемою, оскільки вона включає в себе обчислення відстані кожного примірника тесту з усіма примірниками, що є у тестовій вибірці або в навчальних даних, для обчислення найближчих сусідів; – виконання методу найближчого сусіда в значній мірі залежить від міри відстані, яка визначається між парою екземплярів даних, і яка може ефективно відрізняти нормальні та аномальні. Визначення міри відстані між екземплярами може бути складним завданням, коли дані є складними (графіки, послідовності і т.д.).

1.2.4 Статистичні методи

Розглянемо статистичні методи виявлення аномалій [30].

Обчислювальна складність методів статистичного виявлення аномалій залежить від характеру статистичної моделі, яка потрібна для встановлення

даних. Установка параметричного розподілу від експоненціального сімейства, наприклад, Гауса, Пуассона, поліноміального і т.д., як правило, лінійна за розміром даних, а також кількістю атрибутів. Установка складного розподілу (наприклад, моделі суміші і т.д.) з використанням ітераційних методів оцінки, таких як очікування максимізації (EM-алгоритм), також, як правило, лінійна на кожній ітерації, хоча їх швидкодія може бути низька в залежності від проблеми і / або критерію збіжності. Методи, засновані на ядрах (kernel-based) потенційно можуть мати тимчасову квадратну складність з точки зору розміру даних.

Розглянемо основні переваги та недоліки у таблиці 1.5.

Таблиця 1.5

Переваги та недоліки статистичних методів виявлення аномалій

Статистичні методи виявлення аномалій	
Переваги	**Недоліки**
– якщо припущення щодо базового розподілу даних вірні, статистичні методи забезпечують статистично виправдане рішення для виявлення аномалій; – рейтинг аномалії, отриманий з використанням статистичного методу, пов'язаний з довірчим інтервалом, який може бути використаний в якості додаткової інформації при прийнятті рішень щодо будь-якого екземпляра тесту. Під рейтингом розуміється ймовірність, з якою екземпляр відноситься до аномалії;	– основним недоліком статистичних методів є те, що вони спираються на припущення про те, що дані генеруються з конкретного розподілу. Це припущення часто не має місця, особливо для високих розмірних наборів реальних даних; – навіть коли статистичне припущення може бути досить обґрунтованим, існує кілька статистичних випробувань гіпотез, які можуть бути застосовані для виявлення аномалій; вибирати кращу статистику часто є непростим завданням. Зокрема, побудова гіпотез тестів для складних розподілів, які необхідні, щоб відповідати високим розмірним наборам даних є нетривіальною;

Продовження таблиці 1.5

Статистичні методи виявлення аномалій	
Переваги	**Недоліки**
– якщо крок оцінки розподілу є стійким до аномалії в даних, статистичні методи можуть працювати в неконтрольованій обстановці без будь-якої потреби мічених навчальних даних.	– гістограма на основі вказаних методів відносно проста в реалізації, але ключовий недолік таких методів для багатовимірних даних є те, що вони не в змозі врахувати взаємодію між різними атрибутами. Аномалія може мати значення атрибутів, які окремо часто зустрічаються, але їх поєднання рідко буває, і метод на основі атрибут-мудрої гістограми не зміг би виявити такі аномалії.

1.2.5 Ансамбль методів на прикладі глибинного навчання

Розглянемо таке поняття як ансамбль (мікс) методів на прикладі глибинного навчання. Також розглянемо використання нейронних мереж.

Глибинне навчання (deep learning) – нова галузь досліджень машинного навчання, яка була введена з метою наблизити машинне навчання до однієї з його початкових цілей: штучного інтелекту. У сучасній реальності практично у всьому, що стосується Deep Learning, використовують нейронні мережі. Сучасний Deep Learning здатен упоратися з великими розмірами мереж. А для глибинного навчання використовують спеціальні фреймворки: Keras, Detectron, TensorFlow, PyTorch та інші [37].

Нейронні мережі використовують практично у всіх завданнях, де людина намагається застосувати ШІ. Розглянемо буквально кілька прикладів із посиланнями на більш докладні статті в залежності від типу нейромережі.

Нейронні мережі (neural network, NN) або штучні нейронні мережі (artificial neural networks, ANN) – один із видів машинного навчання. Сьогодні нейронні мережі використовують як альтернативу всім існуючим алгоритмам для машинного перекладу, розпізнавання мови та музики, обробки зображень, визначення об'єктів на фото та відео.

Розглянемо види нейронних мереж та задачі, в яких вони використовуються:

– RNN (рекурентна нейронна мережа) – генерація віршів. Цікавою особливістю даних мереж є їх вміння створювати власні унікальні слова, яких не було в словнику, на якому їх навчали.

– CNN – розпізнавання образів. CNN або Convolutional neural network – одні з найвпливовіших інновацій в області комп'ютерного зору. Застосовуються скрізь, де необхідно розпізнати і/або класифікувати образи/обличчя. Зі складних задач, наприклад, для забезпечення безпеки в аеропортах, на вокзалах, в системах допуску корпорацій із високим рівнем секретності. У щоденному використанні – для більш зручного способу здійснювати покупки.

– GAN – фотореалістичні зображення. GAN або Generative Adversarial Nets використовуються, наприклад, у криміналістиці, коли потрібно створити фоторобот злочинця за описом, в дизайні – для створення предметів одягу або інтер'єру, виходячи з їхнього призначення, в кіновиробництві – для зміни освітлення, настрою кадру, зістарити/омолодити героя. Саме нейронна мережа типу GAN використовується в тестах Фейсбук типу «Як би ти виглядав, якби був протилежної статі».

– DQN – прийняття рішень і боти в іграх. Нейронні мережі типу DQN або Deep Q Learning використовують для прийняття рішень ШІ на підставі аналізу поточної ситуації. Тобто система сама збирає дані, сама їх аналізує, прогнозує найбільш ймовірний результат у тій чи іншій ситуації, приймає максимально вигідне рішення на підставі всіх факторів. Роботу таких нейронних мереж демонструють безпілотні автомобілі, трейдингові боти, чат-боти та ін.

– нейронні мережі (NN) і глибинне навчання (deep learning) – найбільш сучасний підхід до машинного навчання. Нейронні мережі застосовуються там, де потрібні розпізнавання або генерація зображень і відео, складні алгоритми управління або прийняття рішень, машинний переклад і подібні складні завдання [37].

– рекурсивні нейронні мережі (RNN) – аналіз настроїв.

Зазначимо, що глибинне навчання може бути застосоване лише по відношенню до більш складних ANN, що містять кілька прихованих рівнів. При цьому рівні нейронів можуть чергуватися з шарами, які виконують складні логічні перетворення. Кожен наступний рівень мережі шукає взаємозв'язки в попередньому. Така ANN здатна знаходити не тільки прості взаємозв'язки, а й взаємозв'язки між взаємозв'язками. Саме завдяки переходу на нейромережу з глибинним навчанням компанії Google вдалося різко підвищити якість роботи свого популярного продукту «Перекладач». Зокрема, якість перекладу між англійською та французькою мовами підвищилась відразу на 7 балів, тобто більш ніж на 20%.

Нейронна мережа, загалом, – це технологія, побудована для імітації активності людського мозку – зокрема, розпізнавання образів і проходження вхідних даних через різні шари модельованих нейронних зв'язків.

Більш детально розглянувши DNN, зазначимо, що багато експертів визначають глибинні нейронні мережі як мережі, які мають вхідний шар, вихідний шар і хоча б один прихований шар між ними. Кожен шар виконує певні типи сортування та впорядкування в процесі, який деякі називають «ієрархією функцій». Одне з ключових застосувань цих складних нейронних мереж стосується непомічених або неструктурованих даних. Фраза «глибинне навчання» також використовується для опису цих глибоких нейронних мереж, оскільки глибинне навчання є специфічною формою машинного навчання, де технології, використовуючи аспекти штучного інтелекту, прагнуть класифікувати та упорядкувати інформацію способами, які виходять за рамки простих протоколів введення / виведення [75], [76].

На основі вищерозглянутих методів запропонована класифікація існуючих методів виявлення аномалій, що наведена у рис. 1.9.

Недоліки класів існуючих методів виявлення аномалій зведемо до таблиці 1.6.

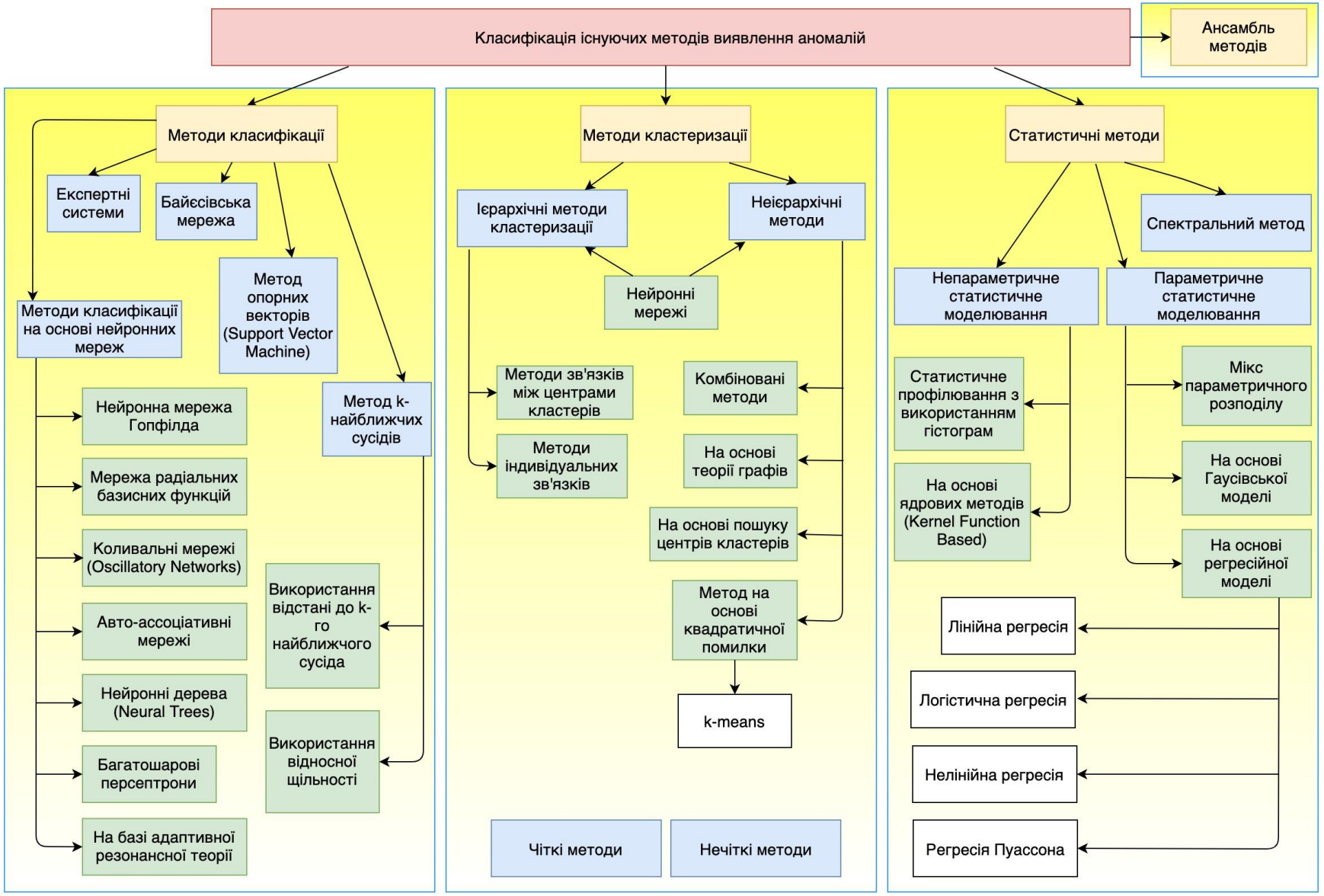


Рисунок 1.9 – Класифікація існуючих методів виявлення аномалій

Таблиця 1.6

Недоліки класів існуючих методів виявлення аномалій

Тип класу методів	Недоліки
Методи класифікації	Недоліком є те, що існуючі методи працюють лише з однорідними даними, а також не дозволяють вказати причину віднесення користувача / набору вхідних даних до певного класу.
Методи кластеризації	
Статистичні методи	
Мікс методів	

Необхідно зазначити, що усі з вище вказаних методів працюють з однорідними даними, тому задача розробки методу подолання різнорідності вхідних даних мобільного додатку залишається актуальною. Тому доцільною є розробка методу виявлення шахрайства як аномалії при інсталюванні мобільних додатків, який на відміну від вище розглянутих, дозволяє аналізувати різнорідні дані для виявлення аномалій в них.

1.3 Аналіз моделей подібності шахрайських шаблонів

Базуючись на вхідних даних та використовуючи параметри органічних користувачів і дані по користувачам, необхідно визначити, в якій мірі користувачі мобільного додатку схожі з еталонними користувачами. Для цього кожен користувач порівнюється з органічними та вираховується коефіцієнт подібності (або оцінка подібності), базуючись на джерелі вченого з Массачусетського технічного інституту Тобі Сегерану [45]. Для цього існує декілька способів, серед яких: евклідова відстань; коефіцієнт кореляції Пірсона; відстань Хеммінга; рангова кореляція Спірмена; косинусна міра подібності; коефіцієнт Танімото.

Так, наприклад, перші два способи також розглядаються в літературі Сегарана, а перших 6 способів визначення подібності визначаються у [77].

Здійснимо аналіз основних характеристик і властивостей даних мір. Зазначимо, що коефіцієнти подібності можна розбити на наступні групи, в залежності від способу їх застосування:

1. для визначення відстані між об’єктами (двома точками у n-вимірному просторі);
2. для визначення статистичного зв’язку між двома змінними;
3. для визначення відмінності між множинами (міри близькості множин).

До першої групи відносяться, так наприклад:

– **Евклідова відстань** – це один з найпростіших способів обчислення оцінки подібності. Евклідовою відстанню називається відстань між двома точками у n-вимірному просторі та визначається формулою [45], [78]:

$$d_{ij}^{(E)} = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}\ , \qquad (1.2)$$

де x_{ik}, x_{jk} – значення k-ї ознаки у об’єктів ω_i и ω_j відповідно;

m – загальна кількість ознак.

– **відстань Хеммінга** (Hamming distance) [81]-[83]. Для визначення міри відмінності між кодовими комбінаціями (двійковими векторами) у векторному просторі кодових послідовностей, відстанню Хеммінга $dist(x, y)$ між двома двійковими послідовностями (векторами) x та y довжини n називається кількість позицій, в яких вони різні. Вілстань Хеммінга можна визначити наступною формулою:

$$d_{ij}^{(H)} = \sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right|\ , \qquad (1.3)$$

де x_{ik}, x_{jk} – значення k-ї ознаки у об’єктів ω_i и ω_j відповідно;

m – загальна кількість ознак.

– **косинусна подібність** [86], [87] або ж косинусна міра подібності – це міра кута між двома ненульовими t -вимірними векторами, які представляють об’єкти у t-вимірному просторі. Два однаково напрямлені вектори мають косинусну міру подібності, рівну «1», два ортогональні вектори мають міру подібності «0», а два вектори, що мають діаметрально протилежні напрями, мають міру подібності «-1». Слід зазначити, що в основному в задачах використовується косинусна міра подібності в діапазоні від «0» до «1».

Одиничні вектори вважаються максимально «подібними», якщо вони паралельні, а у випадку ортогональності вектори вважаються максимально «розбіжними». Косинусна міра подібності обчислюється за такою формулою:

$$sim_1(u,v) = \cos(u,v) = \frac{\langle u,v \rangle}{\|u\| * \|v\|} - \frac{\sum_{k=1}^{t} u_k v_k}{\sqrt{\sum_{k=1}^{t} u_k^2 \sum_{k=1}^{t} v_k^2}}, \quad (1.4)$$

де $u = (u_1,...,u_t)$, $v = (v_1,...,v_t)$ – вектори ознак об’єктів,

u_k, v_k – компоненти (координати) векторів u та v;

$\sum_{k=1}^{t} u_k v_k$ – скалярний добуток між цими векторами.

Тобто з формули (1.4) маємо, що косинусна міра подібності дорівнює відношенню скалярного добутку векторів u та v до добутку норм (довжин) векторів u та v. Зауважимо, чим більше значення дивжин векторів, тим менше буде значення косинусної міри подібності.

Зазначимо, що крім поняття косинусної міри, в задачах класифіквції часто використовується поняття косинусної відстані, яка обчислюється за такою формулою:

$$\dim_1(u,v) = 1 - sim_1(u,v), \tag{1.5}$$

де $sim_1(u,v)$ – косинусна міра подібності.

До другої групи відносяться, так наприклад:

– **коефіцієнт кореляції Пірсона** [45]. Метод Пірсона широко використовується в науці для вимірювання ступеня лінійної залежності між двома змінними, а лінійний кореляційний аналіз дозволяє встановити прямі зв'язки між змінними величинами з їх абсолютними значеннями [80]. Досліджувані ознаки повинні виражатися тільки кількісно. Ця формула дає кращі результати, коли дані погано нормалізовані. Коефіцієнт кореляції обчислюється за формулою:

$$r_{xy} = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 * \sum (y_i - \bar{y})^2}}, \tag{1.6}$$

де x_i – значення, які приймає змінна X;

y_i – значення , які приймає змінна Y;

$\bar{x}$ – середнє значення для X;

$\bar{y}$ – середнє значення для Y.

За допомогою перетворень можна отримати аналог вищезазначеної формули для розрахунку коефіцієнта кореляції [79] :

$$r_{xy} = \frac{n * \sum (x_i * y_i) - \left(\sum x_i * \sum y_i\right)}{\sqrt{\left[n * \sum x_i^2 - (\sum x_i)^2\right] * \left[n * \sum y_i^2 - \left(\sum y_i\right)^2\right]}}, \tag{1.7}$$

де n – кількість спостережень;

x_i – значення, які приймає змінна X;

y_i – значення , які приймає змінна Y.

Коефіцієнт кореляції набуває значень від -1 до 1. Значення +1 означає, що залежність між X та Y є лінійною, і всі точки функції лежать на прямій, яка відображає зростання Y при зростанні X. Значення -1 означає, що всі точки лежать на прямій, яка відображає зменшення Y при зростанні X. Якщо коефіцієнт кореляції Пірсона рівний 0, то саме лінійної кореляції між змінними немає.

– **рангова кореляція Спірмена** [45]. Коефіцієнт рангової кореляції Спірмена дозволяє статистично встановити наявність зв’язку між явищами. Його розрахунок передбачає встановлення для кожної ознаки порядкового номера – рангу. Ранг може бути зростаючим або спадаючим. Для застосування методу Спірмена або рангової кореляції немає жорстких вимог у вираженні ознак – вони можуть бути як кількісними, так і атрибутивними (якісними). Даний метод не встановлює точну силу зв’язку і має орієнтовний характер [84].

Коефіцієнт кореляції рангів відноситься до непараметричних показників зв'язку між змінними, вимірюваними за ранговою шкалою, та підраховується за формулою:

$$\rho = 1 - \frac{6 * \sum\left(D^2\right)}{n * \left(n^2 - 1\right)}, \tag{1.8}$$

де D – різниця між двома рангами кожного спостереження;

$\sum\left(D^2\right)$ – сума квадратів різниць рангів;

n – кількість ранжованих ознак (досліджуваних показників) - число спостережень;

Величина коефіцієнта кореляції Спірмена лежить в інтервалі +1 і -1. Даний коефіцієнт характеризує спрямованість зв'язку між двома ознаками, виміряними ранговою шкалою [79].

– **коефіцієнт рангової кореляції Кендалла** – дещо інший підхід в групі мір зв'язку, базується на підрахунку числа розбіжностей в ранжуванні об’єктів по зіставним змінним x_k и x_j. Існує декілька форм розрахунку коефіцієнта рангової кореляції Кендалла $\tau_{kj}^{(k)}$.

Для незв'язних рангів коэффициент $\tau_{kj}^{(k)}$ розраховується наступним чином:

$$\tau_{kj}^{(k)} = \frac{2(P-Q)}{N(N-1)} \quad \text{або} \quad \tau_{kj}^{(k)} = 1 - \frac{4Q}{N(N-1)}, \tag{1.9}$$

де N – число спостережень;

P – сума чисел, кожне з яких визначається як число наступних за кожним рангом $rang_{ij}$ значень, що перевищують його величину;

Q – сума чисел, кожне з яких визначається як число наступних за кожним рангом $rang_{ij}$ значень, менших його величини.

Якщо в досліджуваній сукупності є зв'язні ранги, то розрахунок коефіцієнта Кендалла $\tau_{kj}^{(k)}$ слід робити наступним чином:

$$\tau_{kj}^{(k)} = \frac{P-Q}{\sqrt{\left(\frac{N(N-1)}{2} - V_k\right)\left(\frac{N(N-1)}{2} - V_j\right)}}, \tag{1.10}$$

де $V_{k/j} = \frac{1}{2}\sum_{l=1}^{p}(t_l^2 - t_l)$, t_l – число повторень l-ї групи рангів в k-й та j-й ранжуваннях, відповідно;

p – кількість груп рангів, що повторюються.

До третьої групи можна віднести так наприклад:

– **коефіцієнт Танімото** [63] – це спосіб визначення міри схожості двох множин. Коефіцієнт Танімото приймає значення від 0 до 1 (чим ближче до 1, тим більше схожість між множинами) та розраховується за такою формулою:

$$T = \frac{N_c}{N_a + N_b - N_c}\ , \tag{1.11}$$

де N_a – кількість елементів в першій множині;

N_b – кількість елементів у другій множині;

N_c – кількість спільних елементів в обох множинах.

1.4 Аналіз сучасних систем виявлення шахрайства при інсталюванні мобільних додатків

Розглянемо найбільш відомі та популярні системи виявлення шахрайства при інсталюванні та відслідковуванні дій мобільних додатків. Алгоритми пошуку шахраїв з більшості систем-аналогів не розкриваються розробниками, проте за отриманими результатами зрозумілі їх основні рішення. Отже, розглянемо відомі характеристики, переваги та недоліки деяких з них у таблиці 1.7.

Слід зазначити, що більшість розглянутих систем, використовуються не лише для виявлення шахрайства при інсталюванні мобільних додатків, а є узагальненими для всіх мобільних транзакцій. Це спричиняє невикористання специфічних даних, які притаманні саме мобільним додаткам, що знижує ефективність виявлення шахрайства при інсталюванні мобільних додатків. Також існує проблема неможливості визначення причини класифікації користувача як шахрайського.

Таблиця 1.7

Відомі характеристики, переваги та недоліки існуючих систем-аналогів виявлення шахрайства при інсталюванні мобільних додатків

Система-аналог	Недоліки
Система Fraudlogix [4]	Має лише «чорний» список IP-адрес, тобто аналізує користувачів лише по одному критерію. Недоліком даної системи є те, що вона не враховує появу нових шахраїв, ботів з новими параметрами та характеристиками та не враховує усі дані користувача.
Kraken [5]	перевіряє належність джерела конверсії до будь-якої з відомих ферм ботів, через що має такий же недолік, як і попередня система. Також дана система не масштабується і робить перевірку на шахрайство лише вибірково по визначеним дням тижня, упускаючи при цьому велику кількість шахраїв.
Adjust [6]	Використовує не всю множину вхідних даних, а перевіряє лише наявність IP-адреси в базі даних VPN, вивчає кількість однакових кліків з одного джерела та порівнює час між подіями. Використання більшої кількості вхідних даних дозволило б більш точно зробити оцінку користувачів та виявити шахраїв.
Kochava [7]	Виявляє шахраїв лише за визначеними критеріями. Критеріїв підібрано достатньо багато, але недоліком такого підходу є те, що у шахраїв з’являються нові характеристики і нові дані, які кожного разу необхідно оновлювати. Також, дана система не використовує усі наявні дані користувача, що знижує її ефективність виявлення шахрайства.
TMC Attribution Analytics [8]	Також використовує не всі вхідні дані користувача та має обмежену кількість критеріїв, за якими визначається шахрайство, через що має такі ж недоліки, як і попередня система.
FraudShield [55]	Відома система виявлення шахрайства FraudShield має досить зручний інтерфейс та можливість налаштування кожного критерію. Проте саме власне налаштування вносить недоліки у дану систему, оскільки через деякі її налаштування можна або пропустити дуже багато шахраїв, або навпаки, через деякі налаштування система вважатиме всіх користувачів шахрайськими.

Продовження таблиці 1.7

Система-аналог	**Недоліки**
Forensiq [29]	Ще одна існуюча система Forensiq є достатньо відомою, кожну конверсію характеризує відповідним рівнем ризику (низький, середній, високий) та перерахуванням підозрілих ознак. Проте неможливо дізнатись чи відстежити, як система визначає рівень ризику та оцінку. Часто ці знання є важливими, особливо у випадках, коли необхідно довести причину визначення користувачів, приведених певною маркетинговою кампанією, як шахрайських при судових позовах.
Appsflyer [10]	Appsflyer є провідною платформою мобільної атрибуції і маркетингової аналітики, яка не так давно випустила власну систему захисту від шахраїв Protect360 [9]. Система має величезну базу IP-адрес та пристроїв, які позначені міткою шахрайства. Як говорить сама компанія, визначена ними мітка шахрайства – це лише привід для додаткової експертизи, оскільки не завжди вона є коректною. Також неможливо дізнатися причину позначення користувача шахраєм, важливість даної можливості вказана у попередній системі.
FraudScore [11]	FraudScore є системою, яка, на відміну від більшості, має оновлення своїх алгоритмів та має самонавчальну систему на базі нейронної мережі. Проте усі свої оцінки вона робить на основі лише деяких даних (таких як дані про пристрій користувача, геодані, дані, пов’язані з операційною системою тощо). Але аналіз не усіх даних також спричиняє невиявлення деяких шахраїв.
AppMetrica [12]	AppMetrica у свою чергу просто провела інтеграцію з попередньо розглянутою системою FraudScore, а отже має ті самі характеристики та недоліки, що і попередня система.

Необхідно відмітити, що одна з відомих компаній – Kochava [7] – має своє рішення для виявлення шахрайства при інсталюванні мобільних додатків [19].

Kochava Data Science аналізує дані для незвичних моделей і визначає норми для виявлення аномальних статистичних витоків. Застосовуючи комбінацію статистичних методологій та алгоритмів ідентифікації шаблонів, процес виявлення шахрайства може ізолювати наступні провідні показники потенційного шахрайства [7]:

1. x_1 – *великі обсяги кліків з однієї IP-адреси.* Багато клієнтів у даний час зосереджують увагу на кількості інсталювань мобільних додатків, але високі обсяги кліків з IP-адрес вносять неясність у результати кампанії. Аномально високі коефіцієнти кліків на інсталювання встановлюють позначку про те, що шахрайство може мати місце. Це також може бути головним (провідним) індикатором проблем із каналами кліків сервер-сервер. У будь-якому випадку проблема вирішується.

2. x_2 – *велика кількість кліків з одного пристрою.* Великий обсяг кліків із одного пристрою не є типовою поведінкою користувача та є значимим індикатором кліків ботів або пристроїв, які викрадали. Як і у випадку з IP-адресами з великими обсягами кліків, Kochava ідентифікує ці ідентифікатори пристроїв до їх анулювання.

3. x_3 – *відхилення середнього часу інсталювання (Mean-time-to-install Outliers - MTTI).* MTTI – це середній час між кліком та встановленням і змінюється залежно від програми та мережі. Наприклад, додатки з картами мають дуже низький MTTI, оскільки завантажуються користувачами, коли їм потрібно кудись прийти. Ігри, як правило, мають більш високі MTTI – користувачі завантажують їх із наміром грати. Коли MTTI сильно варіює для певного джерела - або занадто короткий або довгий, це вказує на схильність до шахрайства.

4. x_4 – *відхилення часу інсталювання (Time-to-install Outliers – TTI).* Цей параметр позначає інсталювання, які відбуваються менш ніж за 30 секунд. Для багатьох додатків на ринку, час інсталювання містить у собі процес натискання реклами, завантаження додатка та його запуск, що менш ніж за 30 секунд дуже малоймовірне. Важливо стежити та вживати заходи, коли велика кількість інсталяцювань із певного джерела демонструє цей шаблон. Цей показник також називають Time Delta.

5. x_5 – *географічні кліки / Дельта при встановленні.* Переважна більшість інсталяцій відбувається в безпосередній близькості до приписаного

кліку. Kochava виявляє статистично значущі відмінності між розташуванням кліка та інсталяції. Ці відхилення можуть бути показниками шахрайства.

6. x_6 – *платформа кліку / Невідповідність інсталяцій.* Якщо оголошення неодноразово подається на неправильній платформі, це може вказувати на шахрайство / аномалію в даних. Якщо реклами для додатка iOS створюють кліки на пристроях, які не належать до операційної системи iOS, це може означати, що бот-ферми генерують шахрайський трафік. Це може також показувати поганий цільовий трафік.

7. x_7 – *відповідність множинних хеш-атрибутів.* Мережі та видавці часто хешують ідентифікатори пристроїв, щоб забезпечити рівень конфіденційності. Проте, коли ідентифікатори хешуються кілька разів, це свідчить про перероблений трафік. Якщо це робиться прозоро, це не викликає занепокоєння. Але якщо трафік вважається ексклюзивним, це може свідчити про шахрайську поведінку.

8. x_8 – *кліки на рекламу*. Ця тактика є однією з простих форм шахрайства для виявлення його на рівні мережі / сайту. Цей параметр вказує, коли одночасно було натиснуто декілька реклам на одному пристрої. Кілька міток часу чітко вказують на накладання кліків / показів або видимість шахрайства.

9. x_9 – *анонімні інсталювання.* Якщо сайти використовують анонімні проксі-сервери, вузли виходу VPN і TOR, щоб приховати свою ідентичність, Kochava додає їх у «чорний» список. Транзакції повинні бути прозорими, а сайт, який вживає заходів, щоб приховати свою ідентичність, не є надійним.

10. x_{10} – *надходження непідтверджених інсталювань.* Для інсталювань, які надходять із iTunes Store, Kochava отримує квитанцію про установку, яку можна перевірити незалежно. У тих випадках, коли інсталяцію неможливо перевірити, Kochava позначає їх як шахрайські.

11. x_{11} – *надходження непідтверджених покупок.* Kochava отримує квитанції про покупки як з iOS, так і з Android. Kochava перевіряє ці покупки на

своїх відповідних серверах-сховищах. Подібно до перевірки квитанцій про інсталювання, Kochava позначає неперевірені покупки як шахрайські.

Kochava є відомою та популярною платформою, але вона визначає лише певні шаблони шахрайства, насправді ж кожного дня шахраї придумують нові способи інсталювання мобільних додатків.

Зважаючи на недоліки вказані в таблиці 1.3, необхідно зазначити, що [16] лише остання пара використовує інтелектуальну складову, серед них AppMetrica просто опирається на FraudScore та використовує їх алгоритми та API, але навіть вказані системи-аналоги виконують рейтингування користувачів на основі не всіх, а вибіркових вхідних даних, тому наявне упущення шахраїв системою [24]. Інші вказані системи використовують відомі бази з шахрайськими даними (наприклад, IP-адресами), що також призводить до упущення шахраїв, що мають інші властивості, шаблони, поведінку.

Очевидно, що причиною вищевказаних недоліків системи є відсутність єдиної концепції виявлення шахрайства на основі всіх наявних даних. Також, недоліком існуючих систем є те, що вони розпізнають лише відомі види шахрайства і не можуть розпізнавати нові шахрайські шаблони, про що також зазначають вчені Хмельницького національного університету [88]. А в сучасному світі важливою є можливість системи адаптуватись, тому необхідним є створення інтелектуальної системи, що матиме змогу самонавчатися.

1.5 Аналіз проблемних аспектів, що виникають при виявлені шахрайства при інсталюванні мобільних додатків та постановка задач дослідження

Необхідно зазначити, що існуючі методи використовують не всю наявну інформацію про користувача. Зокрема, не використовують специфічні дані, які притаманні саме мобільним додаткам, що знижує ефективність виявлення шахрайства при інсталюванні мобільних додатків. Також існує проблема

неможливості визначення причини класифікації користувача як шахрайського існуючими системами.

Тому доцільною є розробка інформаційної технології виявлення шахрайства як аномалії при інсталюванні мобільних додатків, яка на відміну від вище розглянутих, дозволяє аналізувати усю вхідну інформацію про користувача у мобільному додатку для виявлення аномалій в них, а також вказуватиме на причину позначення користувача як шахрая.

Задачі дисертації:

– формалізація процесу виявлення шахрайства як аномалії в даних;

– аналіз та класифікація різнорідних даних при інсталюванні мобільних додатків;

– розробка узагальненого методу виявлення шахрайства при інсталюванні мобільних додатків;

– розробка методу подолання різнорідності вхідних даних;

– розробка інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків.

1.6 Висновки до розділу 1

На основі огляду та аналізу сучасної літератури з галузі виявлення шахрайства в інформаційних технологіях, в роботі запропоновано задачу виявлення шахрайства при інсталюванні мобільних додатків розглядати як задачу пошуку аномалій в даних, тому що шахрайство визначено як навмисне породження аномалії в даних сторонньою особою (шахраєм) або механізмом з певною метою.

Розглянуто та проаналізовано методи і здійснено постановку задачі виявлення шахрайства при інсталюванні мобільних додатків. Проведено варіантний аналіз існуючих методів пошуку аномалій в даних, усі розглянуті методи можна розділити на методи класифікації, методи кластеризації,

статистичні методи, мікс методів на прикладі глибинного навчання. Здійснено аналіз моделей подібності. Показано, що жоден із зазначених методів не може одночасно здійснювати інтелектуальну обробку повної вхідної інформації та вказувати причину, яка дає змогу позначити дані як аномальні.

Виділено такі відомі шахрайські способи під час інсталювання мобільних додатків як кліковий спам (Click Spamming) [1]-[3], мобільне викрадення (Mobile Hijacking) [1], ферми дій (Action Farms) [3]. Проаналізовано існуючі системи виявлення шахрайства при інсталюванні мобільних додатків, серед яких: система Fraudlogix, Kraken, Adjust, Kochava, TMC Attribution Analytics, FraudShield, Forensiq, Appsflyer, FraudScore, AppMetrica. Проте зазначено, що більшість з них видають результат у вигляді таблиці рейтингування, а не чіткої відповіді; опираються на бази з відомими шахрайськими даними (наприклад, з IP-адресами) та не використовують усі важливі вхідні дані. У зв’язку з цим не розпізнаються шахраї, які мають інші властивості, шаблони, поведінку. Також, існує проблема неможливості визначення причини класифікації користувача як шахрайського існуючими системами. Тому й виникла необхідність створення інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків, яка могла б відстежувати та визначати навіть нові шаблони шахраїв і мала змогу самонавчатися.

На основі проведеного аналізу визначено задачі дослідження.

РОЗДІЛ 2 РОЗРОБКА МЕТОДІВ ТА МОДЕЛЕЙ ВИЯВЛЕННЯ ШАХРАЙСТВА ПРИ ІНСТАЛЮВАННІ МОБІЛЬНИХ ДОДАТКІВ З ВИКОРИСТАННЯМ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

Як показано у першому розділі даної роботи, основними передумовами низької точності виявлення шахрайства при інсталюванні мобільних додатків є, насамперед, відсутність методу виявлення шахрайства, який би працював з усіма різнорідними вхідними даними без втрати інформації, а також давав би змогу отримати причину віднесення об’єкту до певного класу.

Тому для підвищення точності процесу виявлення шахрайства, необхідна розробка нового методу виявлення шахрайства, який, на відміну від існуючих, використовуватиме всю повну інформацію про користувачів. Враховуючи різнорідність цієї інформації і складність її аналізу, процес виявлення шахрайства розглядається як пошук аномалій в ній.

2.1 Формалізація процесу виявлення шахрайства як аномалії в даних

Аналіз сучасних підходів до вирішення поставленої задачі, проведений в першому розділі (п. 1.3, 1.4, 1.5), показав, що найчастіше у задачах виявлення аномалій використовуються різнорідні дані. Під різнорідністю, в більшості розглянутих методів та в даній роботі, розуміємо дані різних типів (числові, категоріальні, бінарні) та розмірності, які неможливо рівноцінно порівняти між собою та які приймають значення різних діапазонів. Відомо, що на даний момент у даній області дослідження існує два підходи до зняття різнорідності даних. Перший підхід – це перехід від різнорідних до однорідних даних з використанням сучасних методик, таких як багатовимірне шкалювання [45], one-hot encoding [31]. Використання багатовимірного шкалювання втрачає точність вхідних даних. У свою чергу, використання методу one-hot encoding для

інтерпретації якісної ознаки у вигляді кількісної потребує знання усіх можливих категорій даної ознаки. Проте у випадку з такою ознакою, як, наприклад IP-адреса, неможливо завчасно знати всі можливі категорії. А також збереження матриці усіх можливих значень IP-адрес для кожного користувача займе надто багато ресурсів. Другий – це зменшення кількості даних. До методів другого типу можна віднести метод головних компонент (PCA – Principal Component Analysis [33]). Зменшивши кількість вхідних важливих ознак з використанням методів другого типу, можна отримати неможливість обґрунтування прийнятого системою рішення на основі дійсних вхідних даних. Отже, вхідні дані для виявлення аномалій є різнорідними. Сучасні методи для зняття різнорідності зменшують кількість даних, чим вносять невизначеність в них. Тому є потреба у розробці методу, який зможе «витягти» важливу інформацію з даних та не погіршити при цьому точність результатів, навіть якщо дані є різнорідними.

Таким чином, у процесі виявлення шахрайства при інсталюванні мобільних додатків як аномалії в даних [13]-[14], [28], [60]-[63] важливим є розв’язання задачі аналізу всіх отриманих різнорідних даних, не втрачаючи при цьому точність розв’язання поставленої задачі. У даній роботі запропоновано метод обробки різнорідних даних, який дозволяє вирішити цю задачу.

Для того, щоб вирішити проблеми, які описані в підрозділі 1.5, необхідно формалізувати процес виявлення шахрайства як аномалії в даних при інсталюванні мобільних додатків. Виявлення аномалій відноситься до аналізу даних, а традиційно для аналізу даних використовують теорію множин. Тому і в даній роботі для розв’язання цієї задачі будемо використовувати теорію множин, тому що дана задача відноситься до задачі аналізу даних.

Під час процесу формалізації експертами (Додаток В) було визначено характеристики, за якими може бути виявлено шахрайство в області інсталювання мобільних додатків, що представлено на рис. 2.1.

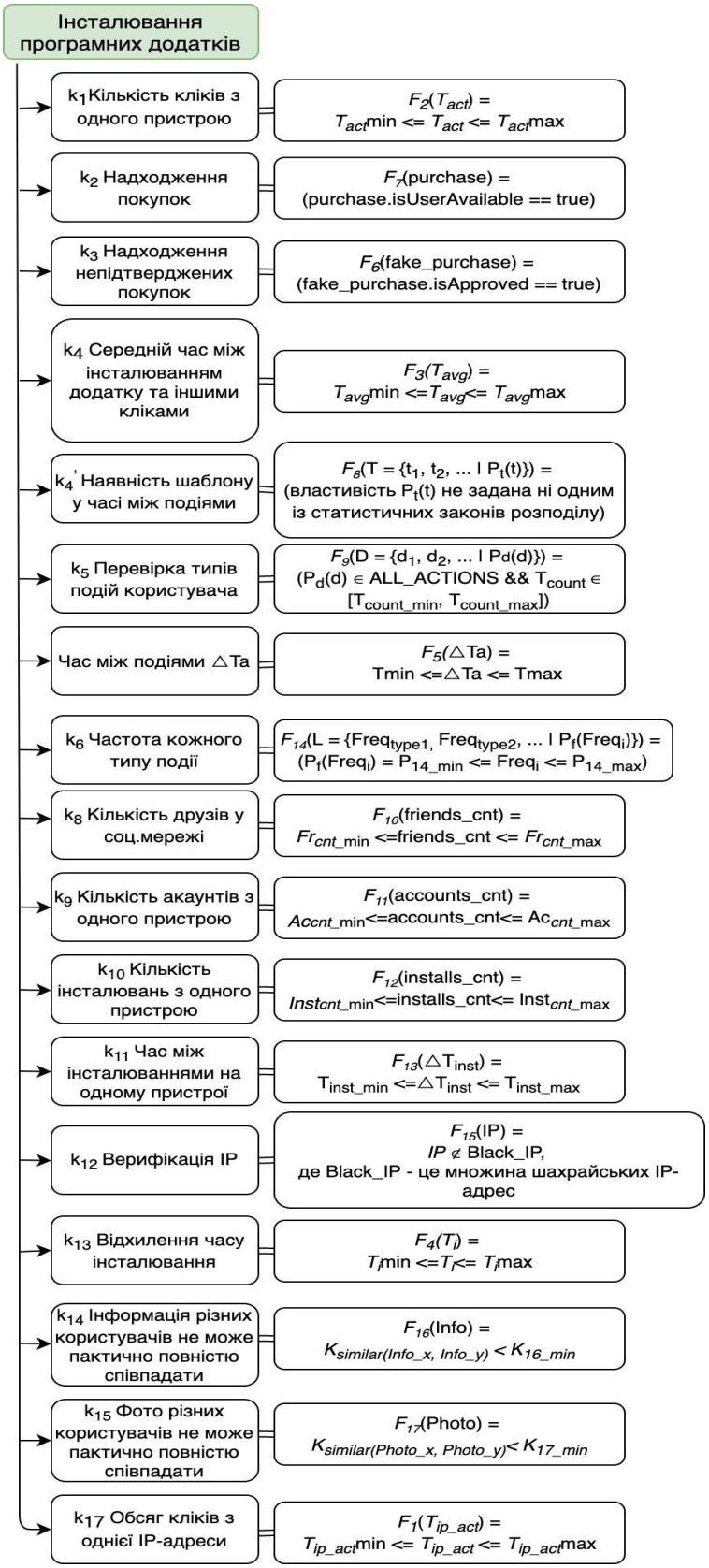


Рисунок 2.1 – Показники та їх значення, за якими може бути виявлено шахрайство при інсталюванні мобільних додатків

На основі інформації, що представлена на рис. 2.1, виділимо вимоги при виявленні шахрайських (аномальних) даних при інсталюванні мобільних додатків.

На думку автора, шахрайськими даними можна вважати ті, які не відповідають хоча б одній із заданих вимог, а саме:

– кількість кліків з однієї IP-адреси за певний проміжок часу або ж інсталювань з однієї IP-адреси T_{ip_act} повинна входити у заданий проміжок. Отже, для цього випадку можна виділити правило $F_1(T_{ip\_act})$, що T_{ip_act} повинен бути меншим або рівним заданого максимуму $T_{ip\_act}\,max$ та більшим або рівним заданого мінімума $T_{ip\_act}\,min$;

– кількість кліків з одного пристрою за певний проміжок часу T_{act} повинна входити у певний проміжок. Отже, для цього випадку можна виділити правило $F_2(T_{act})$, яке визначає, що T_{act} повинен бути меншим або рівним заданого максимуму $T_{ip\_act}\,max$ та більшим або рівним заданого мінімума $T_{ip\_act}\,min$;

– середній час між інсталюванням додатку та іншими діями користувача T_{avg} повинен відповідати певному проміжку. Органічний користувач може робити дії навіть з проміжком у тиждень. Натомість, шахрайські події можуть постійно відбуватися з проміжком у декілька мілісекунд. Тому для цього випадку задамо правило $F_3(T_{avg})$, яке визначає, що T_{avg} повинен бути меншим або рівним заданому мінімальному середньому часу $T_{avg}\,min$ та не перевищувати заданий середній максимум $T_{avg}\,max$;

– час інсталювання T_i також повинен входити у заданий проміжок часу, який задається мінімальним $T_i\,min$ та максимальним $T_i\,max$ часом інсталювання органічного користувача та відповідно правилом $F_4(T_i)$;

– аналогічно до попередніх вимог, час між діями користувача ΔT_a повинен належати заданому проміжку, який задається мінімальним $T\,min$ та максимальним $T\,max$ допустимим значенням проміжку часу та правилом $F_5(\Delta T_a)$ відповідно;

– усі покупки користувача *purchase* повинні бути підтверджені. Якщо користувач надсилає у додаток неіснуючі ідентифікатори покупок, то він намагається зробити шахрайські дії. Дана вимога задається правилом $F_7(purchase)$, яка перевіряє, чи покупка є підтвердженою;

– зворотньою до попередньої є вимога, яка задається правилом $F_6(fake\_purchase)$, де $fake\_purchase$ – це непідтверджені покупки.

Отже, виявлення шахрайства при інсталюванні мобільних додатків можна задати наступним правилом:

$$\mathrm{F} = F_1(T_{ip\_act}) \vee F_2(T_{act}) \vee F_3(T_{avg}) \vee F_4(T_i) \vee F_5(\Delta T_a) \vee F_7(purchase) \vee F_6(fake\_purchase) \vee \ldots \vee F_n(n), \quad (2.1)$$

де *n* – кількість визначених експертами показників (характеристик);

$F_n(n)$ – правило, що визначає значення *n*-го показника.

Для визначення поняття «аномальних даних» визначимо теорему 1.

Означення 1. Множина даних $A = \langle a_1, a_2, ..., a_n \rangle$, що має властивості $P_1(a)$, $P_2(a), \ldots, P_q(a)$, містить неаномальні дані, якщо уся множина даних також має властивості $O_1(a)$, $O_2(a), \ldots, O_t(a)$. Для задачі виявлення аномалій в даних (шахрайства) при інсталюванні мобільних додатків властивості $O_1(a)$, $O_2(a), \ldots,$ $O_t(a)$ задано у вигляді вимог на рис. 2.1.

Теорема 1. Множина даних $A=\langle a_1,a_2,...,a_n\rangle$, що має властивості $P_1(a),P_2(a)...,P_q(a)$, може містити неаномальні дані за визначеними властивостями $O_1(a),O_2(a),...,O_t(a)$, та може містити аномальні дані, які не мають цих властивостей.

Доведення. Нехай є множина даних, яку можна задати у вигляді $A=\langle a_1,a_2,...,a_n\rangle$, що має властивості $P_1(a),P_2(a)...,P_s(a)$, тобто $A=\langle a_1,a_2,...,a_n \mid P_1(a)\ i\ P_2(a)\ i\ P_s(a)\rangle$.

Має місце твердження:

якщо можна виділити додаткові властивості $O_1(a),O_2(a),...,O_t(a)$ неаномальних даних таким чином: $X=\langle a_{org\_1},a_{org\_2},...,a_{org\_n} \mid O_1(a)\ i\ O_2(a)\ i...i\ O_t(a)\ i\ P_1(a)\ i\ P_2(a)\ i...\ P_s(a),a\in A\rangle$, $X\subseteq A$, щоб вони попадали у кластери неаномальних даних $C_{org\_1},C_{org\_2},...,C_{org\_k}$,

то інші дані – $Z\subseteq A$, що не матимуть даних властивостей $O_1(a),O_2(a),...,O_t(a)$, будуть за межами кластерів $C_{org\_1},C_{org\_2},...,C_{org\_k}$ та вважатимуться аномальними (рис. 2.2).

З цього слідує, що $X\cap Z=\varnothing$. Тобто усіх користувачів множини даних A можна однозначно поділити лише на два класи:

– неаномальні – ті, що входять у кластери $C_{org\_1},C_{org\_2},...,C_{org\_k}$ та мають властивості $O_1(a),O_2(a),...,O_t(a)$,

– аномальні – ті, що не входять у кластери $C_{org\_1},C_{org\_2},...,C_{org\_k}$ та не мають властивості $O_1(a),O_2(a),...,O_t(a)$.

На базі визначеної теореми 1, множина аномальних даних Z розглядається у роботі як група (множина) даних $\langle a_1,a_2,...,a_q\rangle$, яка входить у множину вхідних даних $A=\langle a_1,a_2,...,a_n\rangle$, що характеризується множиною властивостей

$P_1(a), P_2(a)..., P_s(a)$, але виходять за задані межі властивостей $O_1(a), O_2(a), ..., O_t(a)$, що притаманні неаномальним даним X множини вхідних даних A, де $X \subseteq A, Z \subseteq A$ та, згідно теореми, $X \cap Z = \varnothing$. У задачі виявлення аномалій у вхідних наборах даних з мобільних додатків, аномальними будемо вважати ті дані (елементи множини), які [16]:

– не мають властивостей $O_1(a), O_2(a), ..., O_t(a)$, що визначають множину неаномальних даних X;

– не співпадають по розмірності;

– не співпадають по властивостям групи даних;

– не входять в область гранично допустимих значень множини неаномальних даних X – задамо це властивістю $O_L(a)$, яка має вигляд $(min\_value \le x \le max\_value)$.

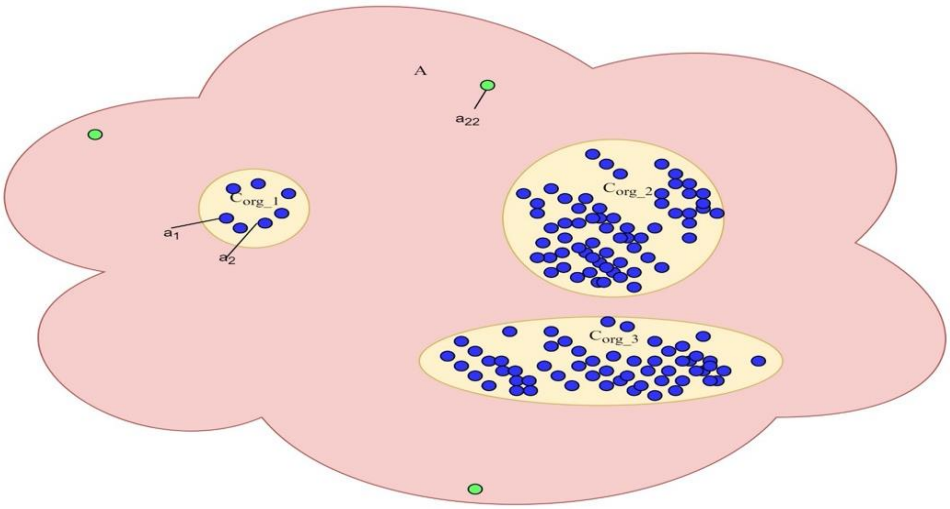


Рисунок 2.2 – Множина даних A, що поділяється на лише неаномальні дані, які поділені по кластерам $C_{org\_1}, C_{org\_2}, C_{org\_3}$, та аномальні дані, які не потрапляють у кластери $C_{org\_1}, C_{org\_2}, C_{org\_3}$

Отже, подамо математично вхідний набір даних у формулі 2.2.

$$A = \{a \mid P_1(a)\ i\ P_2(a)\ i\ ...i\ P_s(a)\}, \qquad (2.2)$$

де $P_1(a), P_2(a)..., P_s(a)$ – це властивості множини A.

Розглянемо нижче варіанти вхідних наборів даних найбільш поширених систем прийняття рішень та систем штучного інтелекту, в яких наявні аномалії [17]. Дані варіанти також розглянуто в роботі [108].

1. Один з можливих варіантів, у якому чітко видно аномалію – це наявність у вхідному наборі A групи елементів, які не мають властивостей набору A, проте всі мають одну спільну властивість, а отже, чітко виділяються на фоні інших даних (формула 2.3).

$$A = \{x_1, x_2, ...x_n, ..., z_1, z_2, ..., z_k, ... \mid x \in X\ i\ z \in Z\}, \qquad (2.3)$$

де $x_1, x_2, ...x_n$ – елементи підмножини

$X = \{x_1, x_2, ...x_n \mid O_1(a)\ i\ O_2(a)\ i...i\ O_t(a)\}$

мають властивості $O_1(a)\ i\ O_2(a)\ i...i\ O_t(a)$ та є неаномальними даними;

$z_1, z_2, ..., z_k$ – елементи підмножини $Z = \{z_1, z_2, ..., z_k, ... \mid P_a(z)\}$, у свою чергу, не мають цих властивостей, і це означає, що вони є аномальними у заданій множині даних, та всі мають спільну властивість $P_y(a)$, завдяки якій вони виражено групуються в окремий клас, відповідно.

$$X \subseteq A \text{ та } Z \subseteq A \text{ та } X \bigcap Z = \varnothing. \qquad (2.4)$$

Більш наочно цей випадок показано на рис. 2.3 з використанням діаграми Венна, приклад цієї ж аномалії у двовимірному просторі – на рис. 2.4.

2. Ще один з можливих варіантів, у якому чітко видно аномалію – це наявність у вхідному наборі *A* групи елементів, які не мають властивостей набору *A*, і всі, у свою чергу, мають різні властивості (формула 2.5).

$$A = \{x_1, z_1, x_2, z_2, ..., x_n, ..., z_k, ... \mid x \in X \ i \ z \in Z\}, \quad (2.5)$$

де $x_1, x_2, ...x_n$ – елементи підмножини

$X = \{x_1, x_2, ...x_n, ... \mid P_1(x) \ i \ P_2(x) \ i ... i \ P_s(x)\}$ мають властивості

$P_1(x) \ i \ P_2(x) \ i ... i \ P_s(x)$ та є не аномальними даними;

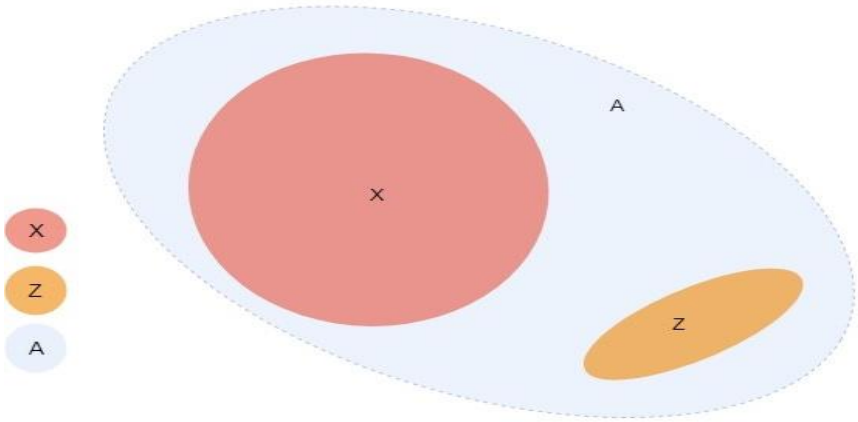


Рисунок 2.3 – Діаграма Венна, де X – підмножина неаномальних даних, Z – підмножина аномальних даних

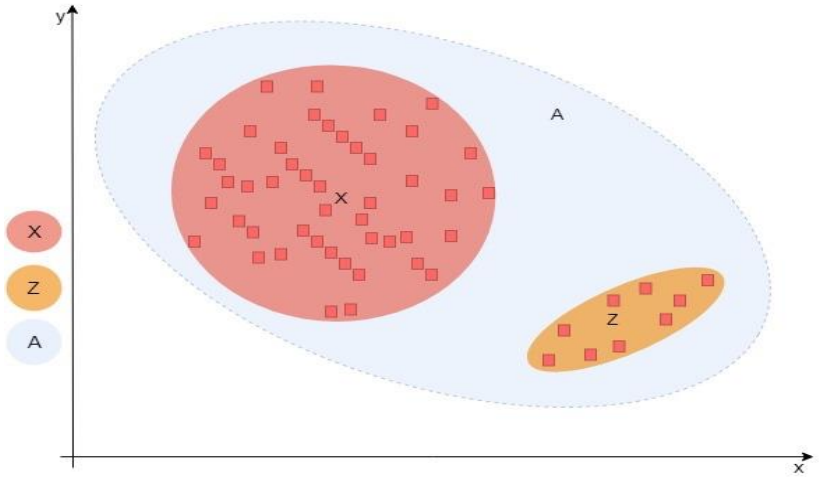


Рисунок 2.4 – Приклад аномалії у двовимірному просторі, де X – підмножина неаномальних даних, Z – підмножина аномальних даних

$z_1, z_2, ..., z_k$ – елементи підмножини

$Z = \{z_1, z_2, ..., z_k, ... | P_{a1}(z)$ *або* $P_{a2}(z)$ *або... або* $P_{k1}(z)\}$, у свою чергу, не мають цих властивостей, що означає, що вони є аномальними у заданій множині даних *A*; та всі мають різні властивості – одну з $P_{a1}(z)$ *або* $P_{a2}(z)$ *або... або* $P_{k1}(z)$, тому вони хаотично розкидані, відповідно $X \subseteq A$ та $Z \subseteq A$ та $X \bigcap Z = \varnothing$.

Більш наглядно цей випадок показано на рис. 2.5 з використанням діаграми Венна та приклад цієї ж аномалії у двовимірному просторі на рис. 2.6.

На рис. 2.5, 2.6 подано випадок, в якому множина вхідних елементів має властивості, що обмежують елементи множини областю гранично допустимих значень. Так, наприклад, для множини *X* властивості $P_1(x), P_2(x), ..., P_s(x)$ можуть мати наступний вигляд:

$$P_1(x) = x \geq 100000,$$
$$P_2(x) = x \neq 205$$
$$\ldots,$$
$$P_s(x) = x < 3001 \ i \ x > 25. \qquad (2.6)$$

Представимо у двовимірному просторі випадок, який задається областями гранично допустимих значень на рис. 2.7, де сукупність підмножин $\{X_1, X_2\}$ є покриттям підмножини неаномальних даних *X*, *Z* – підмножина аномальних даних.

3. Варіант, у якому спостерігається аномалія – це наявність у вхідному наборі А групи елементів, які не співпадають по властивостям групи даних.

Для більшого розуміння даного варіанту зобразимо діаграму Венна на рис. 2.8, де всі елементи підмножин X_1, X_2, X_3, X_4, X_5 є неаномальними, проте

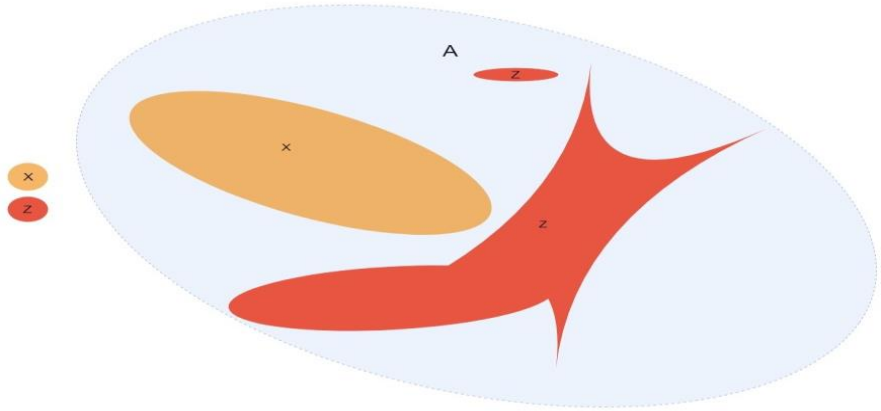


Рисунок 2.5 – Діаграма Венна, де X – підмножина неаномальних даних, Z – підмножина аномальних даних

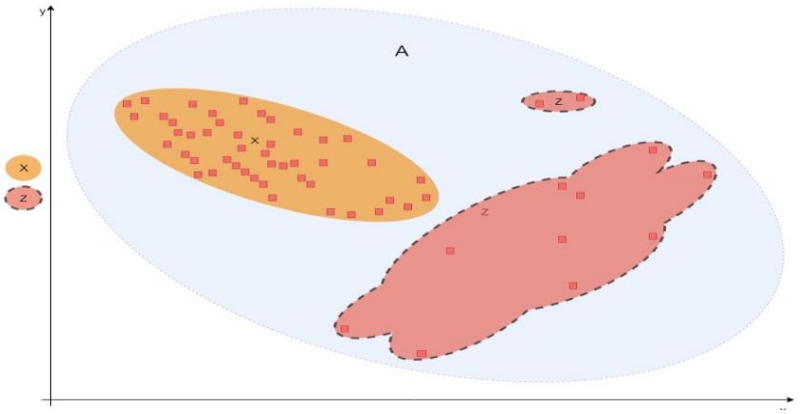


Рисунок 2.6 – Приклад аномалії у двовимірному просторі, де X – підмножина неаномальних даних, Z – підмножина аномальних даних

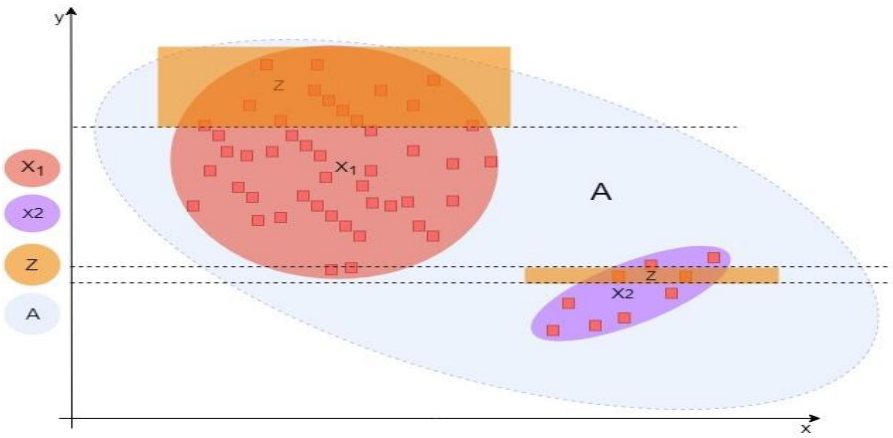


Рисунок 2.7 – Приклад аномалії у двовимірному просторі, в якому неаномальні дані задаються областями гранично допустимих значень

деякі об’єднання цих підмножин мають властивості множини аномальних даних *Z*. Формалізуємо даний вираз.

Отже, *A* – множина вхідних даних, яка зображає даний випадок. Проте для початку задамо множину неаномальних даних *X* (формула 2.7).

$$X = \{x_1, x_2, ...x_n, ... | P_1(x)\ i\ P_2(x)\ i...i\ P_s(x)\}, \quad (2.7)$$

у свою чергу сукупність підмножин $\{ X_1, X_2, ..., X_5 \}$ є покриттям множини *X*.

Також задамо множину аномальних даних *Z* у формулі 2.8.

$$Z = \{x_1 x_2 ... x_n ... | x_1, x_2, ..., x_n \in X\ ma\ P_{a1}(z)\ або\ P_{a2}(z)\ або...\ або\ P_{k1}(z)\}. \quad (2.8)$$

Тоді множину вхідних даних *A* можна задати формулою 2.9.

$$A = \{x_1 x_2 ... x_n ... | x_1, x_2, ..., x_n \in X\ i\ x_1 x_2 ... x_n ... \notin Z\}. \quad (2.9)$$

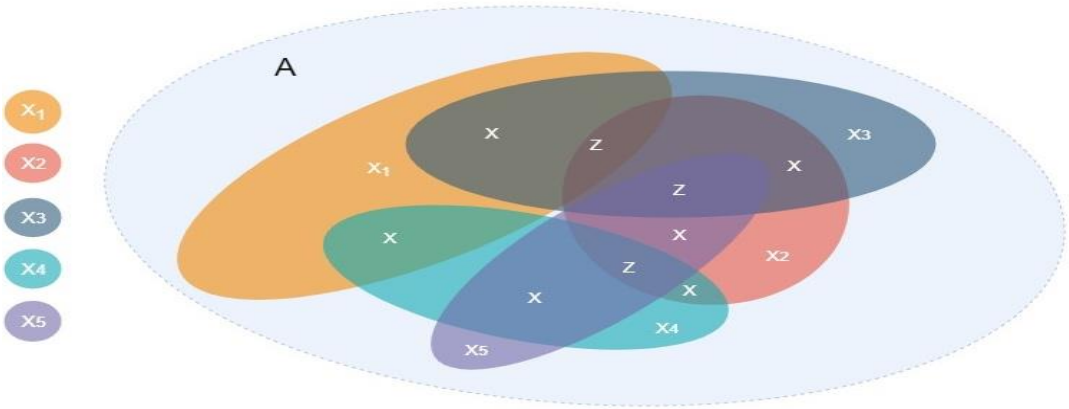


Рисунок 2.8 – Діаграма Венна, де всі елементи підмножин X_1, X_2, X_3, X_4, X_5 є неаномальними, *Z* – множина аномальних даних

4. Варіант, у якому спостерігається аномалія – це наявність у вхідному наборі A підмножини неаномальних даних X, яка представляє собою об’єднання

підмножин $X_1, X_2, ..., X_5$. Проте деякі з підмножин X_i перевищують допустиму розмірність, або ж їх розмірність менша, ніж це можливо в неаномальних даних.

Математично даний випадок можна представити знову ж сукупністю підмножин $\{X_1, X_2, ..., X_s\}$, які є покриттям підмножини неаномальних даних *X*, яка входить у множину вхідних даних *A*. У свою чергу, деякі з підмножин $\{X_1, X_2, ..., X_s\}$ мають задані властивості розмірності, як представлено у формулі 2.10.

$$X_1 = \{x \mid count(x) < K\}, \tag{2.10}$$

де $count(x)$ – кількість елементів у множині X_1;

K – задане значення.

У свою чергу, множина *A*

$$A = \{a \mid a \in X\}. \tag{2.11}$$

Приклад даної аномалії представлено у двовимірному просторі, в якому дані обмежуються розмірністю множини на рис. 2.9. Так, випадок шахрайства, як одного із типів аномалії, можна спостерігати при інсталюванні мобільних додатків тоді, коли у конкретній локації живе, наприклад, 300000 жителів, а за один день з цієї локації пройшло 1000000 інсталювань. Тоді множина X_1, яка характеризуватиметься обмеженням (2.12) з одного пристрою за один день, вважатиметься аномальною, оскільки значно перевищуватиме задане обмеження розмірності.

$$count(x) < 300000 + \varepsilon\ , \tag{2.12}$$

де $count(x)$ - це кількість інсталювань з певної локації;

ε - це допустиме відхилення.

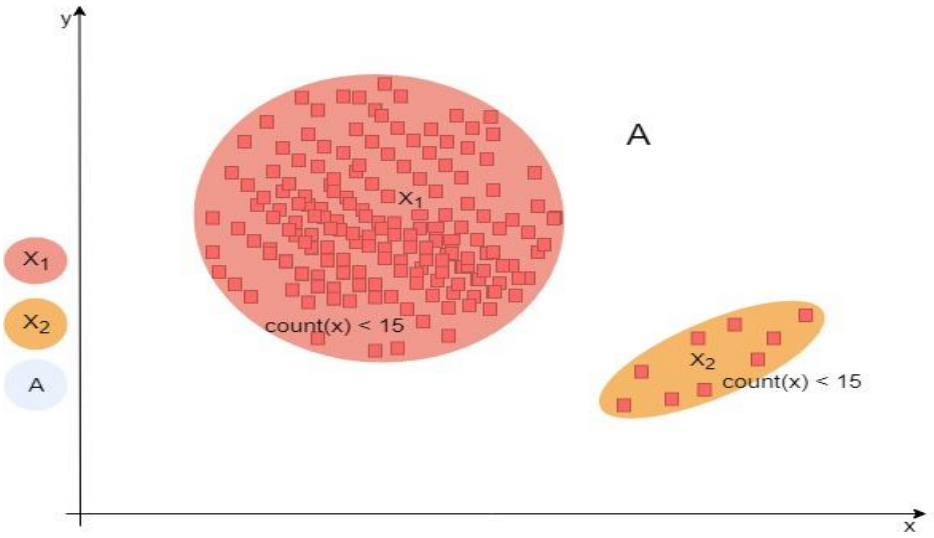


Рисунок 2.9 – Приклад аномалії у двовимірному просторі, в якому дані обмежуються розмірністю множини

Розглянувши приклади аномалій у вхідних наборах даних, формалізуємо дані підрозділу 1.1 та представимо вхідні дані множиною, яка має властивості, наведені у підрозділі 1.1 та на рис. 2.1. Отже, формалізуємо вхідні дані інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків. Матимемо множину користувачів

$$S = \{a_1, a_2, ..., a_w \mid a \in A\}, \tag{2.13}$$

в якій вхідні дані по кожному з користувачів задаються множиною A

$$A = \{x_1, z_1, x_2, z_2, ..., x_n, ..., z_k, ... \mid x \in X \ i \ z \in Z\}, \tag{2.14}$$

де елементи підмножини $X = \{ x_1, x_2, ..., x_n, ... \mid P_1(x) \ i \ P_2(x) \ i \ ... \ i \ P_s(x)\}$

мають властивості $P_1(x)\ i\ P_2(x)\ i\ ...\ i\ P_s(x)$ та є неаномальними даними; елементи підмножини $Z = \{ z_1, z_2, ..., z_k, ... \mid P_{a1}(z)$ або $P_{a2}(z)$ або…або $P_{ak}(z)\}$ не мають цих властивостей, тобто вони є аномальними у заданій множині даних A, а також мають свої задані властивості $P_{a1}(z)$ або $P_{a2}(z)$ або … або $P_{ak}(z)$, відповідно $X \subseteq A\ ma\ Z \subseteq A\ ma\ X \cap Z = \varnothing$.

У свою чергу, якщо множина даних користувача міститиме елементи підмножини *Z*, то користувач вважатиметься шахраєм, інакше – органічним.

Таким чином, у даному підрозділі формалізовано процес виявлення шахрайства як аномалій в даних з використанням теорії множин. Для формалізації процесу виявлення шахрайства як аномалії в даних у поставленій в даній роботі задачі та для розробки методу виявлення шахрайства при інсталюванні мобільних додатків, для початку, необхідно здійснити аналіз та класифікацію різнорідних вхідних даних у предметній області, що розглядається.

Для формалізації процесу виявлення шахрайства у визначеній предметній області – при інсталюванні мобільних додатків – та для вибору, із множини розглянутих у підрозділі 1.3, методу для розв’язання поставленої задачі, важливим є аналіз вхідних даних, з якими працюють розглянуті методи, та даних, які відомі для розв’язання поставленої задачі. Тому розглянемо аналіз та класифікацію різнорідних даних при інсталюванні мобільних додатків.

2.2 Аналіз та класифікація різнорідних даних при інсталюванні мобільних додатків

Для здійснення класифікації даних та визначення множини вхідних даних, які необхідні при прийнятті рішень про наявність шахрайства, розглянемо детально весь процес появи та зміни даних [14]. У процесі інсталювання мобільного додатку поведінка кожного користувача, який приймає в цьому участь, характеризується набором подій, наприклад: інсталювання додатку,

підтвердження інсталювання, відкриття додатку, реєстрація тощо. Послідовна множина можливих вхідних даних представлена на рис. 2.10.

З рис. 2.10 видно можливу послідовність надходження подій про кожного з користувачів, поділених на множини *L1, L2, L3* в залежності від типу події. Відзначимо, що спільними ознаками кожної події є:

– інформація про поточного користувача, а саме унікальний ідентифікатор (*userID*);

– інформація про поточний пристрій, а саме його унікальний ідентифікатор (*user deviceID*), IP-адреса (*user IP*), інформація про операційну систему пристрою (*user deviceOS*);

– інформація про поточну подію, а саме її тип (*event type*) та час (*event time*).

Також кожна з подій може містити ознаки, необхідні лише для неї, так наприклад:

– кожна внутрішня подія мобільного додатку містить кортеж специфічної для себе інформації (*eventSpecificData*);

– подія отримання підтвердження про інсталювання додатку містить прапорець *isApproved*, який позначає, чи інсталювання підтверджене, чи ні;

– подія про реєстрацію користувача містить інформацію, яку користувач ввів про себе (*user information*), його фотографію (*user photo*) та дату народження (*user date of birth*);

– подія про покупку містить унікальний ідентифікатор купленого товару (*offerID*) та покупки (*purchaseID*), ціну покупки (*purchasePrice*) та прапорець *isApproved* про підтвердження або непідтвердження покупки.

У процесі дослідження даної роботи всю множину вхідних даних, яка необхідна для виявлення шахрайства при інсталюванні мобільних додатків, було розділено на декілька множин:

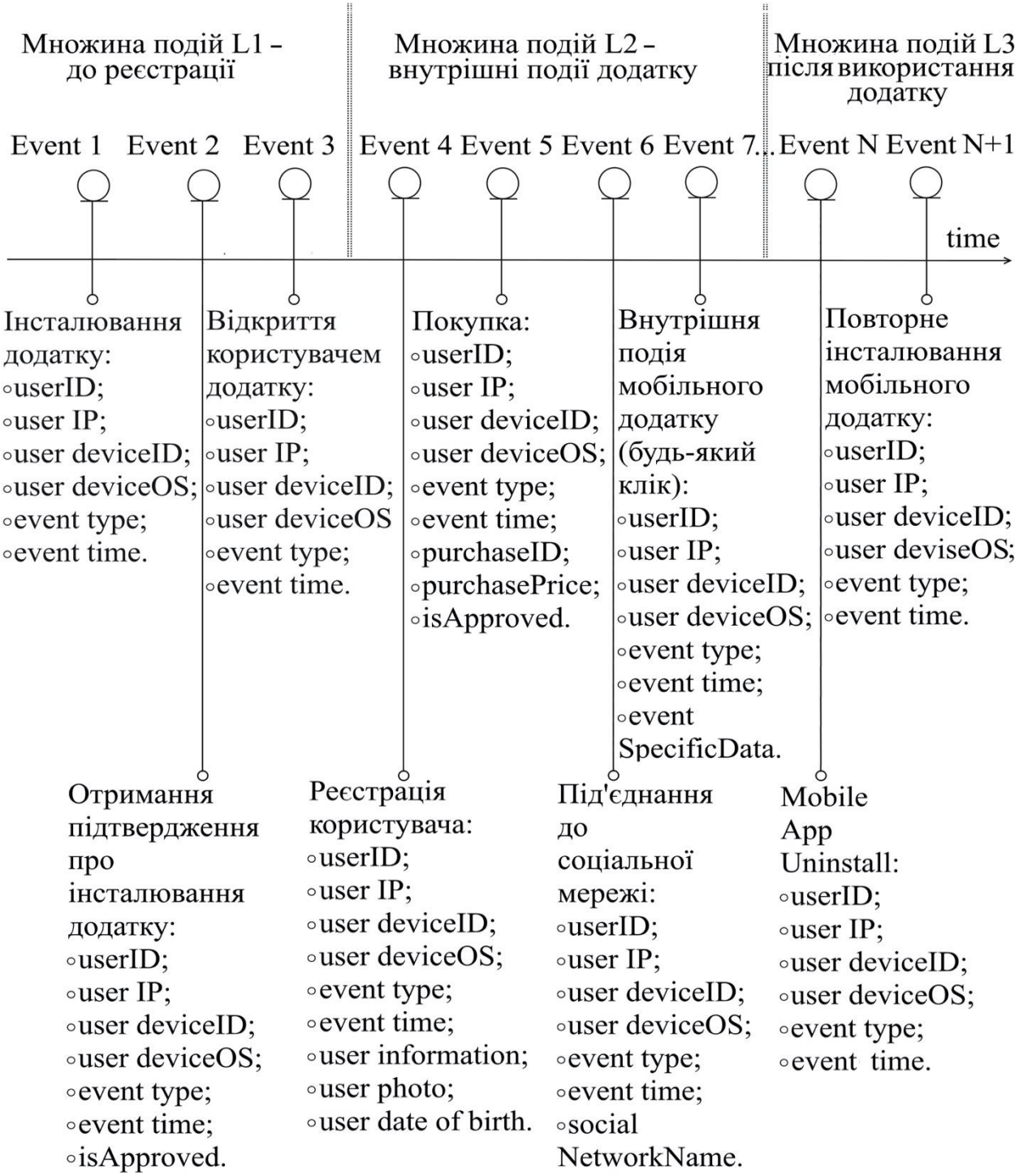


Рисунок 2.10. Послідовність надходження подій про кожного з користувачів при інсталюванні мобільних додатків

– множина *M1*: дані та інформація про користувача під час інсталювання;

– множина *M2*: інформація про користувача та його дії після інсталювання;

– множина *M3*: дані про користувача під час деінсталювання.

Таке розділення даних на підмножини в подальшому дозволить розробити процедуру виявлення аномалій (шахраїв) при інсталюванні мобільних додатків.

На рис. 2.11 представлено класифікацію вхідних даних за їх типом.

Представлений аналіз даних показав, що дані в таких системах є різнорідними, а саме – різних шаблонів, метрик, розмірностей. З рис. 2.10 та 2.11 видно, що серед вхідних даних є як якісні та кількісні, так і масиви якісних і кількісних даних, тому зрозуміло, що аналіз даних в даному контексті – це дуже складна задача.

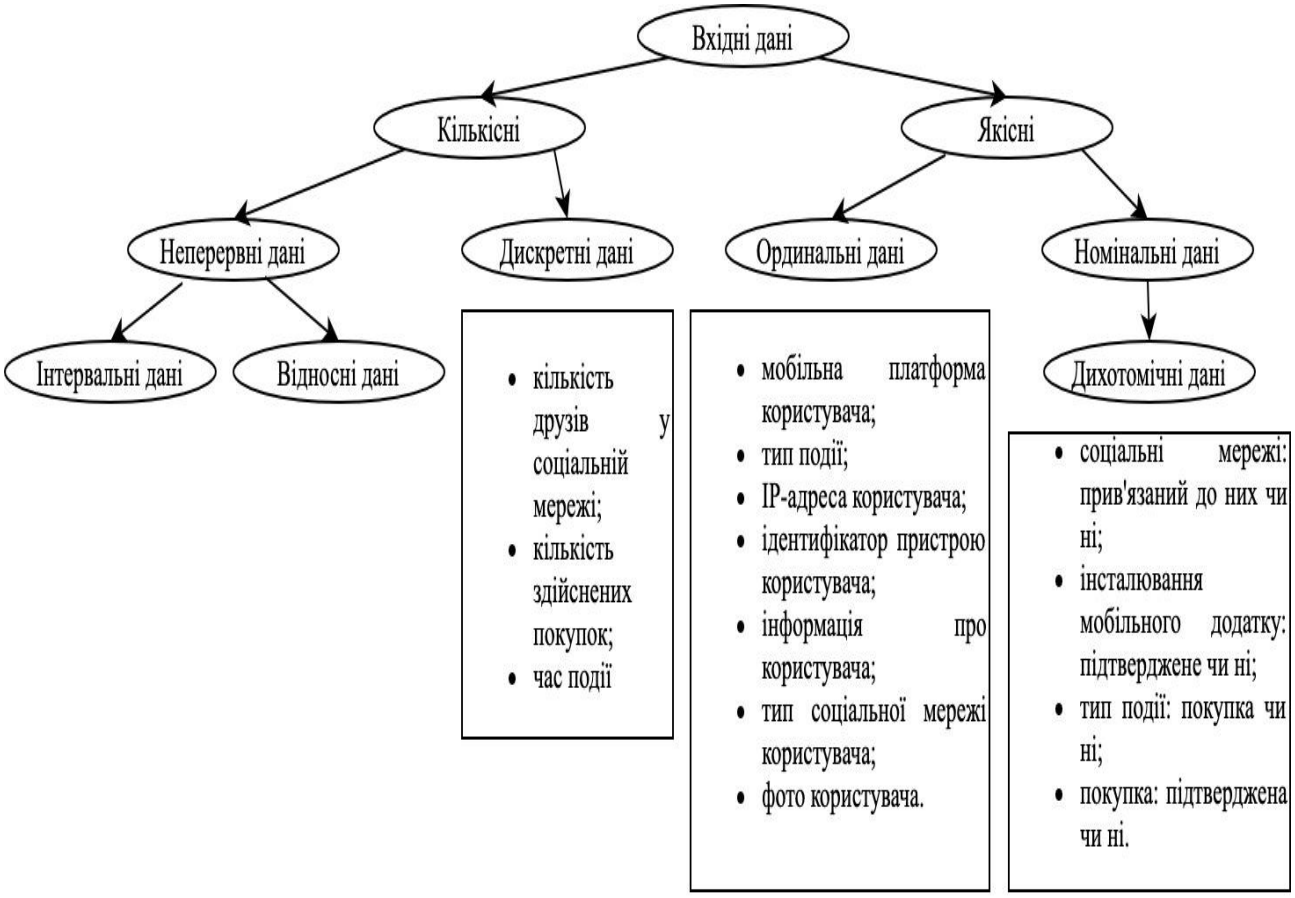


Рисунок 2.11. Класифікація різнорідних вхідних даних в системах виявлення шахрайства при інсталюванні мобільних додатків

2.3 Розробка узагальненого методу виявлення шахрайства при інсталюванні мобільних додатків

Отже, для аналізу таких даних необхідно застосувати метод, який працює з різнорідними даними. Проте, як було розглянуто в підрозділі 1.2, один з найпоширеніших у наш час підходів вирішення схожих задач – нейронні та нейроподібні мережі [51] – також працюють лише з кількісними даними. Наразі, існує підхід переведення категоріальних даних у кількісні – one-hot encoding [31], але для застосування даного підходу необхідно знати всю множину можливих значень категоріальної ознаки. Проте, наприклад, для такої ознаки як IP-адреса неможливо завчасно знати всі можливі її варіанти. Навіть типи подій постійно будуть додаватися у ході удосконалення (improvement-а) мобільного додатку, що призводитиме до постійного зростання кількості вхідних ознак, даних по них та до постійного перенавчання системи. У випадку з IP-адресами, буде спостерігатись надто стрімке збільшення вхідних категорій, на зберігання яких по кожному користувачу буде необхідна велика кількість ресурсів. Взагалі, у більшості сучасних алгоритмів практично не підтримується робота з категоріальними даними [14], [15].

Зазначимо, що ще однією значною проблемою для методу, що розробляється у даному досліджені, є неможливість використання нейронних мереж для подолання проблеми різнорідності даних. Причина полягає у тому, що на даний момент неможливо чітко обґрунтувати рішення, прийняте нейронними мережами, що є важливим при розгляді судових позовів, в яких необхідно чітко наводити аргументи. Так, наприклад, це одна з найвагоміших причин, із-за якої безпілотні автомобілі ще не доступні у продажу, а в досліджуваній області таке обґрунтування є обов’язковим.

Ще один метод, який працює і з кількісними, і з якісними даними – узагальнений дискримінантний аналіз [41], [60] – також потребує визначену множину можливих значень категоріальної змінної, аналогічно до one-hot

encoding. А в нашому випадку не всі категоріальні вхідні дані мають дискретну множину категорій.

При розгляді інших існуючих методів нормалізації даних, потрібно відмітити, що вони не працюють з кількісними і якісними даними, тому може спостерігатись зменшення точності через втрату інформації та допущені похибки. Серед розглянутих методів зазначимо методи переведення даних від різнорідних до однорідних (наприклад, багатовимірне шкалювання) або методи виділення найважливіших ознак (наприклад, Principal Component Analysis [33]).

Більшість сучасних методів працює з однорідними даними, тому виникла необхідність розробки методу, який використовує отримані у процесі виявлення шахрайства різнорідні дані. Це є важливою задачею. Зважаючи на поставлену проблему, при розробці узагальненого методу виявлення шахрайства при інсталюванні мобільних додатків необхідно вирішити наступні задачі [14]:

– подолання різнорідності вхідних даних з метою використання усієї інформації про користувачів при класифікації користувачів та для підвищення точності результатів;

– класифікація користувачів на органічних та шахрайських на основі данних, отриманих після подолання їх різнорідності;

– створення узагальненого портрету шахрая.

Всі задачі можна об’єднати в єдиний метод, а саме – в узагальнений метод, який складатиметься з наступних етапів (рис. 2.11):

– виявлення характеристик даних користувача;

– подолання різнорідності даних;

– побудова моделі класифікації;

– класифікація;

– формування бази знань для виявлення шахрайства;

– формування бази даних шахраїв;

– інтелектуальний аналіз наявних даних та формування портрету користувача;

– створення узагальненого портрету шахрая та прогнозування.

При побудові інформаційної технології наявність шахраїв в роботі розглядається як наявність аномалії у великих масивах різнорідних даних про користувачів. Базуючись на поставлених задачах, розглянемо кожну з них.

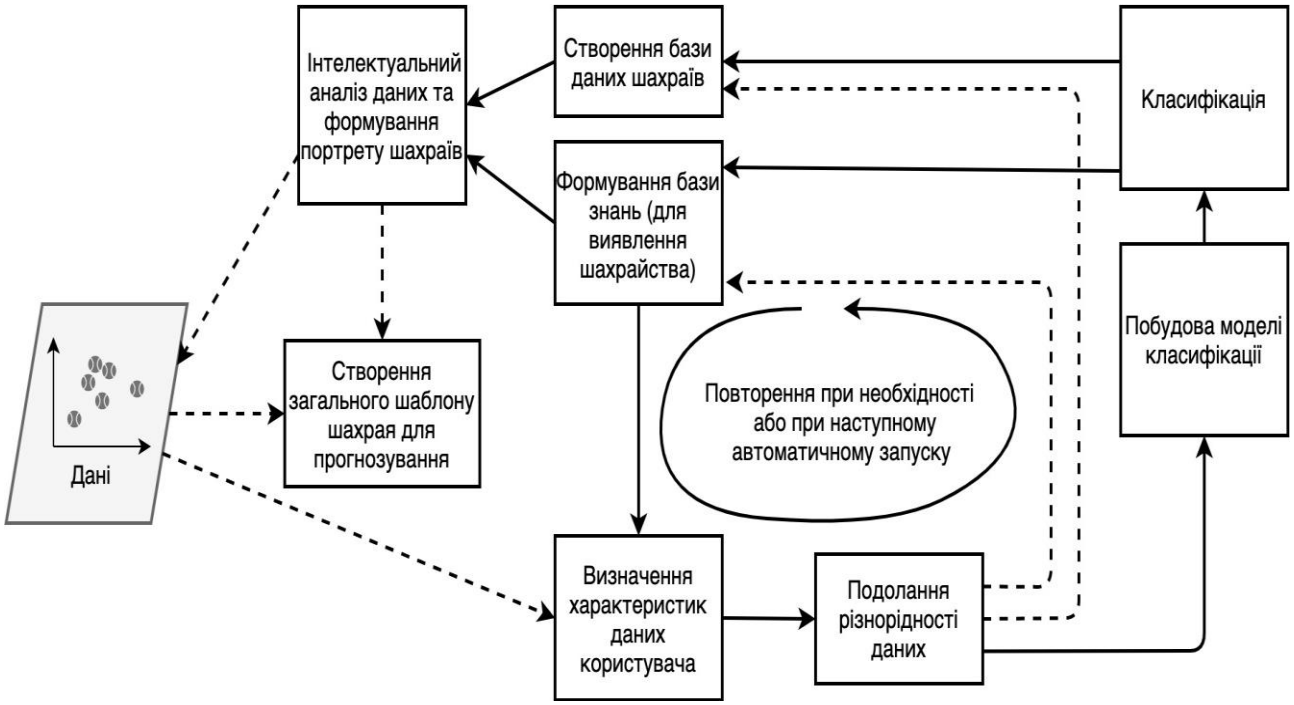


Рисунок 2.12. Етапи узагальненого методу виявлення шахрайства при інсталюванні мобільних додатків

2.3.1 Розробка методу подолання різнорідності вхідних даних

Для розробки методу подолання різнорідності вхідних даних, з метою використання усієї інформації про користувачів для підвищення точності результатів та можливістю їх використання при класифікації користувачів, для початку, необхідно класифікувати вхідні дані про користувачів. Слід зазначити, що метод подолання різнорідності вхідних даних являє собою сукупність процедур вибору ознак, зниження розмірності та нормалізації даних.

2.3.1.1 Класифікація вхідних даних, їх шкалювання за інформативністю та формалізація

Оскільки всі дані є різнорідними [12], як було зазначено вище (рис. 2.11), працювати з такими даними складно і багато існуючих систем, методів та моделей відкидають, наприклад, масиви якісних даних чи дані, всі значення яких завчасно невідомі тощо. Так, наприклад, кожен з користувачів мобільного додатку має такий якісний параметр як IP-адресу – це і є однією з характеристик, якою часто нехтують. Зазначимо, що деякі з систем перевіряють наявність поточної IP-адреси в існуючій базі даних з шахрайськими IP-адресами, але не подають її у якості ознаки (feature) в систему інтелектуального аналізу даних. Це відбувається через ряд причин: моделі інтелектуального аналізу даних, в основному, працюють лише з числовими значеннями; для перетворення якісних даних у кількісні зазвичай використовується метод one-hot encoding [31], але для його використання необхідна завчасно відома множина можливих значень ознак (feature), у випадку з IP-адресою завчасно невідома вся множина IP-адрес майбутніх користувачів мобільного додатку. При цьому можливо згенерувати всі можливі значення IP-адрес версій протоколів IPv4 та IPv6, проте це призведе до зберігання величезної кількості надлишкової інформації, що значно зменшить швидкодію та погіршить ефективність роботи системи. Тому зазвичай такі дані відкидаються і не відіграють роль у процесі прийняття кінцевого рішення – користувач шахрай чи ні. Але чим більша кількість вхідних даних, тим більше залежностей, кореляцій можна з них виділити і більше різних шаблонів шахраїв можна буде помітити. Тобто, чим більше даних для прийняття рішень, тим вища точність системи. Тому у роботі постало питання як правильно та ефективно використовувати усі різнорідні дані про користувача, оскільки практично всі системи класифікації працюють з однорідними даними.

Зазвичай в різних областях науки і техніки для аналізу таких даних використовують різні методи шкалювання, так, наприклад [16]:

– системи правил, побудованих з використанням нечіткої логіки;

– нормалізацію даних, що використовується, наприклад, у задачах статистичної обробки даних;

– звичайним шкалювання вимірювальних пристроїв.

Проте, як можна для даних про користувачів при інсталюванні мобільних додатків, що представлені на рис. 2.9, 2.10, провести шкалювання? Відомо, що IP-адреса може мати вигляд, наприклад, такий як «127.0.0.1», «192.168.5.1». Нормалізувати дані такого типу неможливо. Аналогічно недоцільно використовувати систему правил для даних такого типу. Оскільки для того, щоб взяти до уваги всі ці дані, необхідно сформувати 4–12 правил, що суттєво ускладнює процес аналізу. Отже, серед вказаних трьох варіантів залишається шкалювання. Проте, на думку автора, при аналізі таких даних традиційне шкалювання всього набору даних не є можливим, оскільки значення у всіх цих даних різне (число, категорія якісних даних, масиви кількісних чи якісних даних), а інформація, яку вони несуть, однозначно визначає чи дана ознака характеризує шахрая, чи має властивості органічного користувача. Оскільки важливим є здійснити не лише рейтингування користувача, але й зазначити причину, чому даного користувача помітили шахраєм, тому в роботі [14] був запропонований метод подолання різнорідності даних про користувача по цінності інформації, яку вони несуть.

Отже, для подолання різнорідності даних використовується шкалювання за інформативністю – по цінності інформації, яку несуть ці дані, – що означає перетворення всіх даних (якісних і кількісних) до єдиної нормалізованої шкали зі значеннями від 0 до 1 в залежності не від значення, а від інформації, яку несуть ці дані. Наприклад, розглянемо шкалу для переведення ознаки «IP-адреса» до бінарного значення, представлену на рис. 2.13. Після шкалювання за даною ознакою, деяких користувачів можна однозначно віднести до шахраїв. Вважатимемо, що значення 0 на шкалі означатиме, що користувач є шахраєм за даною ознакою, а значення 1 на шкалі означатиме, що користувач є органічним

(тобто не є шахраєм) за даною ознакою. Це є особливістю запропонованого методу шкалювання різнорідних даних.

Однак, при використанні такого шкалювання не всіх користувачів можна однозначно віднести або до шахраїв, або до органічних, оскільки ця IP-адреса може буде відсутня в базі даних. Тому для таких користувачів необхідно додатково використати інтелектуальну обробку даних. Для цього, для початку, необхідно визначити вхідні дані та їх характеристики.

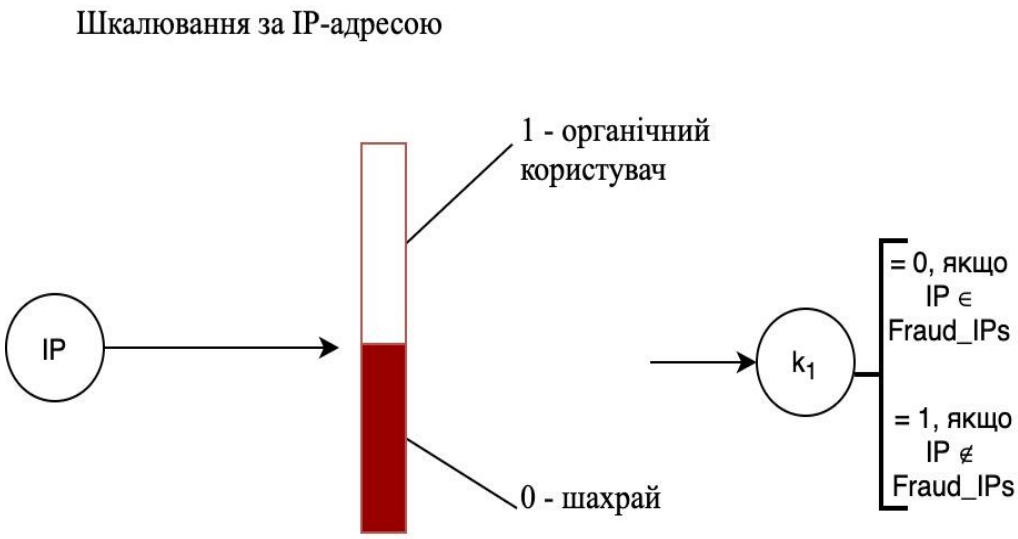


Рисунок 2.13. Шкалювання за IP-адресою

Визначення кількості найбільш важливих вхідних даних та-характеристик цих даних в процесі дослідження виявилось досить складною задачею. Тому, для визначення повного масиву вхідних даних та характеристик цих даних, в роботі було проведено експертне опитування (Додаток В), в якому прийняли участь 13 експертів провідних IT-компаній України, Швейцарії, США, Нідерландів, які мають досвід виявлення шахрайства.

Метою експертного опитування було визначити таку множину вхідних даних та характеристик даних, за допомогою яких можна визначити, чи користувач є шахраєм, чи ні (є органічним користувачем). Для цього опитування було проведено у два етапи. На першому етапі експертам було надано набір усіх

можливих вхідних даних, які необхідно проранжувати або додати інші вхідні дані. На другому етапі експертами для визначеної множини вхідних даних були визначені граничні значення. Так, наприклад, визначено:

– граничні значення кількості кліків з однієї IP-адреси Tip_act_min та $Tip\_act\_\max$;

– граничні значення часу між подіями інсталювання додатку та реєстрації користувача $Tinst\_\min$ та $Tinst\_\max$;

– мінімальна та максимальна кількість друзів у соціальних мережах $Cf\_\min$ та $Cf\_\max$.

Результати першого етапу експертного опитування представлено на рис. 2.14. Як видно з визначених за допомогою експертного опитування вхідних даних, дані є різнорідними, містять і якісні, і кількісні показники та приймають значення з різних діапазонів. Класифікація різнорідних даних представлена на рис. 2.14.

Результати другого етапу експертного оцінювання представлено з використанням теорії множин, а саме у вигляді компоненту

$$O = (D, X), \tag{2.15}$$

де D – множина подій по кожному користувачу;

X – характеристики даних, визначені експертами, що представлені у вигляді властивостей множини даних D.

Дані властивості повинні бути притаманні усім органічним користувачам. Розглянемо їх.

Властивість (2.16) перевірятиме, чи IP-адреса є шахрайською чи ні. Можна

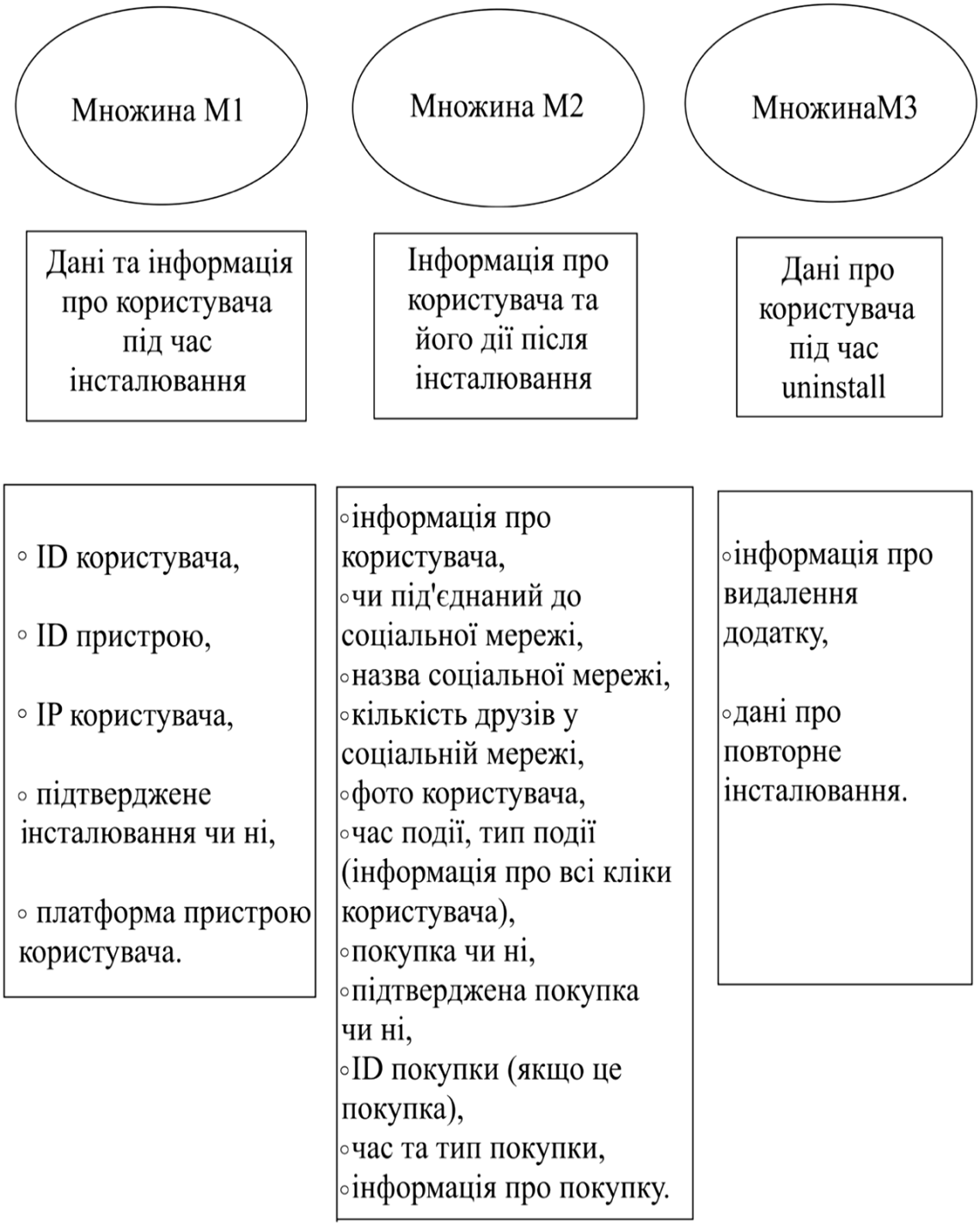


Рисунок 2.14. Вхідні дані розділені на три виділені множини

зазначити, що при $IP \in FRAUD\_IP$ користувач, який має дану IP-адресу, точно є шахраєм.

$$P_1(IP) = IP \notin FRAUD\_IP, \quad (2.16)$$

де IP – IP-адреса, яку використовував користувач;

$FRAUD\_IP$ – множина відомих шахрайських IP-адрес.

Властивість (2.17) перевірятиме, чи ID – унікальний ідентифікатор користувача є шахрайським чи ні. Аналогічно до попередньої властивості, можна зазначити, що при $ID \in FRAUD\_ID$ даний користувач є шахраєм.

$$P_2(ID) = ID \notin FRAUD\_ID, \quad (2.17)$$

де ID – унікальний ідентифікатор користувача;

$FRAUD\_ID$ – множина ідентифікаторів відомих шахраїв.

Властивість (2.18) перевірятиме унікальний ідентифікатор пристрою (мобільного телефону). Слід зазначити, що за умови $D\_ID \in FRAUD\_D\_ID$ даний користувач є шахраєм.

$$P_3(D\_ID) = D\_ID \notin FRAUD\_D\_ID \quad (2.18)$$

де D_ID – унікальний ідентифікатор пристрою (мобільного телефону);

$FRAUD\_D\_ID$ – множина ідентифікаторів пристроїв, з яких були помічені шахраї.

Властивість (2.19) перевіряє ідентифікатор покупки. Можна зазначити, що якщо у користувача є хоч одна непідтверджена покупка, то даний користувач однозначно є шахраєм.

$$P_4(P_{id}) = (P_{id}\, isAproved = true), \quad (2.19)$$

де P_{id} – це унікальний ідентифікатор покупки, яку зробив користувач;

isAproved – прапорець, який позначає, чи покупка підтверджена відповідним магазином мобільної платформи чи ні.

Властивість (2.20) перевіряє типи подій, доступних користувачу на даному етапі. Можна зазначити, що при $T \notin AVAILABLE\_TYPES$ даний користувач є шахраєм.

$$P_5(T) = (T \in AVAILABLE\_TYPES), \quad (2.20)$$

де T – тип поточної події, яка доступна користувачу на даному етапі.;

AVAILABLE _TYPES – множина типів подій, доступна користувачу на даному етапі.

За умови, якщо властивість (2.21) не виконується, то користувачі, що використовують пристрій з поточним ідентифікатором *deviceID* позначаються шахраями.

$$P_6(D\_A_{cnt}) = D\_A_{cnt} \le 5, \quad (2.21)$$

де D_A_{cnt} – кількість акаунтів на одному пристрої (device).

Аналогічно до попередньої властивості, якщо умова (2.22) не виконується, то користувачі, які користуються даною IP-адресою вважаються шахраями.

$$P_7(IP\_A_{cnt}) = IP\_A_{cnt} \le 5, \quad (2.22)$$

де IP_A_{cnt} – кількість акаунтів з однієї IP-адреси.

Для опису наступних властивостей виділимо з множини D підмножини подій по кожному з пристроїв користувача (по $deviceID$) $E_1, E_2, ..., E_d$, де d – це кількість пристроїв, на які користувач встановлював мобільні додатки, тобто $E_1 \cup E_2 \cup ... \cup E_d = E$. Варто зазначити, що $E_1 \subseteq E$, $E_2 \subseteq E$, …, $E_d \subseteq E$ та $E_1 \bigcap E_2 \bigcap ... \bigcap E_d = \varnothing$. У свою чергу, кожну з виділених підмножин можна поділити на підмножини дій користувача по кожній хвилині, а саме:

$$E_{1,1} \cup E_{1,2} \cup ... \cup E_{1,m1} = E_1, E_{1,1} \bigcap E_{1,2} \bigcap ... \bigcap E_{1,m1} = \varnothing,$$
$$E_{2,1} \cup E_{2,2} \cup ... \cup E_{2,m2} = E_2, E_{2,1} \bigcap E_{2,2} \bigcap ... \bigcap E_{2,m2} = \varnothing,$$
$$\ldots,$$
$$E_{d,1} \cup E_{d,2} \cup ... \cup E_{d,md} = E_d, E_{d,1} \bigcap E_{d,2} \bigcap ... \bigcap E_{d,md} = \varnothing, \quad (2.23)$$

де $m1, m2, ...md$ – кількість хвилин, проведених у додатку з пристрою $1,2,...d$ відповідно.

Тоді кожну з підмножин $E_{1,1}, ..., E_{1,m1}, E_{2,1}, ..., E_{2,m2}, E_{d,1}, ..., E_{d,md}$ можна задати наступним чином:

$$\{e \mid P_{e1}(c), P_{e1}(c)\} = c \le 50, \quad (2.24)$$

де c – кількість елементів даної підмножини.

Дана властивість позначає, що органічний користувач може робити не більше, ніж 50 кліків (подій) у хвилину. Якщо ж подій значно більше, то користувача можна вважати підозрілим. Для того, щоб однозначно визначити, чи користувач є шахраєм за даною властивістю – а саме за часом його подій, необхідно перевіривти ряд наступних властивостей. Для цього представимо підмножини $E_1, E_2, ..., E_d$ у наступному вигляді:

$$\{d \mid P_{d1}(t_{di}, t_{di+1}), P_{d2}(t), P_{d3}(n), P_{d4}(t_{din}, t_{do})\}, \quad (2.25)$$
$$P_{d1}(t_{di}, t_{di+1}) = t_{di+1} - t_{di} \geq 60000mc,$$

де t_{di}, t_{di+1} – час між сусідніми подіями.

Дана властивість перевіряє час між подіями, і якщо час більший визначеного, то користувач вважається підозрілим та перевіряються наступні властивості.

$$P_{d2}(t) = E_t \notin F, \quad (2.26)$$

де t – час подій;

E_t – множина часу подій користувача;

F – множина повністю відомих законів розподілу.

Дана властивість (2.26) перевіряє належність множини часу подій користувача деякому повністю відомому закону розподілу. Така перевірка є необхідною через те, що зазвичай шахрайські скрипти використовують random-функції для вибору часу між подіями. Але будь-який «random» побудований на основі певного відомого закону розподілу, зазвичай нормального закону розподілу.

$$P_{d3}(n) = E_n \notin F, \quad (2.27)$$

де n – тип (назва) подій;

E_n – множина часу подій користувача;

F – множина повністю відомих законів розподілу.

Дана властивість (2.27) аналогічна до попередньої, і перевіряє розподілення типу події. Така перевірка так само пояснюється тим, що шахрайські скрипти, які дозволяють привести навіть мільярди інсталювань за добу/годину, так само

вибирають тип події, яку виконуватиме «користувач» (тобто скрипт), з використанням random-функції, яка основана на певному законі розподілу.

$$P_{d4}\left(t_{d\_1}, t_{d\_2}\right) = t_{d\_2} - t_{d\_1} \geq 120000 \text{ мс} \qquad (2.28)$$

де $t_{d\_2} - t_{d\_1}$ – час між сусідніми подіями.

Аналогічні властивості будуть мати підмножини поділені не за $deviceID$, а за $userIP$.

Звичайно ж, на основі другого етапу експертного опитування визначено основні характеристики даних, проте з кожним певним проміжком часу можуть з'являтися нові шахрайські способи інсталювання додатків, саме тому при розробці методу також буде застосовуватись і інтелектуальний аналіз даних (data mining). Адже, згідно визначення Жерона Орельєна [31], data mining – це застосування прийомів машинного навчання для дослідження великих об'ємів даних (Big Data) та подальшого виявлення шаблонів, які не були помічені відразу.

Базуючись на пункті 1.1.1 та вищерозглянутих формулах 2.15–2.25, формалізуємо вхідні дані у межах задачі виявлення аномалії як шахрайства при інсталюванні мобільних додатків. У таблиці 2.1 представлено типи вхідних даних та їх формалізація з використанням теорії множин.

Для подолання різнорідності даних, авторами використовується шкалювання (нормалізація) даних, що означає перетворення всіх даних (якісних і кількісних) до єдиної шкали від 0 до 1. Значення 0 на шкалі означатиме, що користувач по даній ознаці є шахраєм, а значення 1 на шкалі означатиме, що користувач по даній ознаці є органічним. Це є особливістю запропонованої шкали. Так, наприклад, [14], [16], [18]:

– IP-адреса – якісні дані, уся множина значень яких невідома, переводиться у коефіцієнт, значення якого 0 або 1 (рис. 2.15);

– тип події – якісні дані, в яких, на відміну від попередньої ознаки, уся множина типів подій завчасно відома;

– наявність непідтвердженої покупки – якісна ознака, а саме, дихотомічна. При наявності непідтвердженої покупки даний коефіцієнт рівний 0, що позначає користувача шахраєм. Інакше, даний коефіцієнт рівний 1;

– множина часу подій користувача переводиться у значення від 0 до 1, де 0 означає, що множина повністю належить певному закону розподілу, а 1 – не належить (рис. 2.16);

– кількість друзів у соціальній мережі – це кількісні дані, які також переводяться у коефіцієнт із значенням від 0 до 1. Якщо кількість друзів входить у граничні значення, визначені на другому етапі експертного оцінювання, то значення коефіцієнту рівне 1, інакше – 0. Приклад кількісної ознаки подано на рис. 2.17.

Як видно з прикладів, показаних на рис. 2.15–2.17, виконується перехід від різнорідних до однорідних даних, за допомогою коефіцієнтів. Таке перетворення різнорідних даних в однорідні в подальшому для виявлення шахрая дає змогу використати методи класифікації.

В процесі шкалювання та на основі результатів експертного оцінювання в роботі було розроблено 17 шкал. Для конвертації наявних даних до значень на розроблених шкалах, у роботі було сформовано 17 коефіцієнтів, які впливають на прийняття рішень при виявленні шахрайства. Аналіз цих коефіцієнтів дозволив здійснити поділ коефіцієнтів на такі групи:

а) **перша група** охоплює коефіцієнти, які дозволяють провести попередній аналіз даних, а саме, однозначно з множини користувачів визначити шахрайських користувачів, органічних та підозрілих. Так, наприклад, у першу групу входить 13 коефіцієнтів. Розглянемо деякі з них:

Таблиця 2.1

Типи вхідних даних та їх формалізація

Тип вхідних даних по кожному з користувачів окремо	**Формалізація вхідних даних**
k_1– Кількість кліків з одного пристрою за хвилину	$X_1 = \{x_1, x_2, ...x_n \mid P_1(x)\}$, де $P_1(x) = x \le T_{avg}$, T_{avg} – середня кількість кліків органічного користувача за хвилину. Оскільки немає чітко визначеної середньої кількості кліків органічного користувача за хвилину, то даний коефіцієнт k_1 необхідно порівнювати з узагальненим шаблоном шахрая та визначати подібність у межах від 0 до 1, де $k_1 = 0$ означатиме, що користувач – шахрай, а $k_1 = 1$ – означатиме, що користувач точно не є шахраєм. Для цього візьмемо середній показник по відомим органічним користувачам T_{avg} та визначимо відсоткову частку від нього. Отже, $T_{avg} = \frac{\frac{T_1}{T_{1avg}} + \frac{T_2}{T_{2avg}} + ... + \frac{T_N}{T_{Navg}}}{N}$, де N – кількість хвилин, які користувач зареєстрований у мобільному додатку, $T_1, T_2, ..., T_N$ – це кількість кліків користувача у хвилину 1, 2, …, N відповідно, $T_{1avg}, T_{2avg}, ..., T_{Navg}$ – це середня кількість кліків по відомим органічним користувачам у хвилину 1, 2, …, N відповідно. У свою чергу, $k_1 = \frac{\sum_1^n x_i}{T_{avg}}$

Продовження таблиці 2.1

Тип вхідних даних по кожному з користувачів окремо	**Формалізація вхідних даних**
Покупки: k_2 - Куплена / не куплена	Даний показник присутній не у всіх користувачів. Звичайно ж, кожна компанія-розробник прагне, щоб всі користувачі робили покупки у їхньому додатку (якщо у додатку є така функціональність), проте, базуючись на науці геміфікації [98], завжди є група людей, яка не робитиме покупки через психологічні упередження / установки.
	k_2 – Можна вважати, що людина, яка зробила хоч одну покупку, точно є не шахраєм. Проте, нічого не заважає шахраю зробити одну покупку, щоб потім заробити більші кошти, ніж він витратив. Тому будемо робити перевірку по декільком правилам: якщо кількість зроблених користувачем покупок $\geq K_{2\_min}$, то k_2 попередньо рівне 1 та означає, що користувач не є шахраєм, якщо кількість покупок < K_{2_min}, то k_2 попередньо рівне 0.5, що означає, що користувач може бути як шахраєм, так і органічним користувачем. Коефіцієнт K_{2_min} у даній роботі визначається на основі експериментально-експертного опитування (в експертному опитуванні взяли участь провідні програмісти декількох фірм (Google, Microsoft, Onseo, Plexteq тощо)) (Додаток В), яке вноситься як початкове значення по даному коефіцієнту при формуванні початкового узагальненого шаблону шахрая та коригується у ході корекції узагальненого шаблону шахрая; також, користувач міг купити покупку, яка ще недоступна для нього. У цьому випадку виникає запитання про те, як він дізнався про її існування (або це помилка системи, або ж шахрайство). Тому введемо $B = \{b_1, b_2, ..., b_n\}$ – множина доступних користувачу покупок. І перевірятимемо вхідний вектор з покупками на належність їх до множини B $X_2 = \{x \mid x \in B\}$. При спробі користувачем зробити недоступну йому покупку, коефіцієнт k_2 автоматично стає рівним 0, що означає, що користувач є шахраєм.

Продовження таблиці 2.1

Тип вхідних даних по кожному з користувачів окремо	**Формалізація вхідних даних**
Покупки: k_3 – Підтверджена / не підтверджена	k_3 - якщо є непідтверджена покупка, то користувача автоматично можна вважати шахраєм і $k_3 = 0$, інакше – $k_3 = 1$.
k_4 – Час між подіями користувача	Визначається схоже до коефіцієнта k_1, а саме: $k_4 = \frac{\sum_1^p x_i}{R_{avg}}$, де $X_4 = \{x_1, x_2, ..., x_p\}$ – вектор дельт часу між подіями, $R_{avg} = \frac{R_{1avg} + R_{2avg} + ... + R_{Uavg}}{U}$, де U – кількість відомих органічних користувачів, $R_{1avg}, R_{2avg}, ..., R_{Uavg}$ – це середня дельта часу між подіями по відомим органічним користувачам: першому, другому, …, U-ому відповідно. Оскільки дане значення, так само як і k_1, немає чітко визначеного значення, тому воно аналогічно визначається з використанням коефіцієнту подібності.
k_4' – Час між подіями користувача (перевірка на бота) за певний проміжок часу	k_4' – буде показувати, чи є залежність між часом подій користувача, яку можна задати ймовірнісними законами. Адже всі скрипти рандомізовані, тобто час між подіями задається, наприклад, нормальним розподілом / розподілом Бернуллі тощо. Даний коефіцієнт буде рівний відношенню всіх дій користувача до тих дій користувача, які задані певним ймовірнісним законом. Належність вибірки до закону розподілу можна визначити з використанням критерію Колмогорова-Смірнова [99].

Продовження таблиці 2.1

Тип вхідних даних по кожному з користувачів окремо	**Формалізація вхідних даних**
k_5 – Тип подій користувача – скільки доступних подій він використовує	Спочатку збирається інформація по типам подій користувача, коли користувач стає постійним (користується мобільним додатком хоча б протягом K_{5_min} – кількість днів, що визначається на основі експериментально-експертного оцінювання (Додаток В)), визначаємо, якщо користувач робить незвичні для себе події, то це може бути шахрайством. Як відомо, навіть постійних користувачів «зламують» та користуються їх акаунтами. Отже, тип подій користувача можна задати множиною $Y = \{y_1, y_2, ..., y_j \mid y \in E\}$, де $E = \{e_1, e_2, ..., e_w\}$ – вектор подій користувача, який користується додатком не менше 1 місяця. Також k_5 визначатиметься за наступною формулою: $k_5 = \frac{U_C}{c}$, де U_C – кількість типів подій, які робить даний користувач; C – загальна кількість типів подій, які доступні користувачу. При $k_5 \in \lfloor K_{5\_min\_cnt}; K_{5\_max\_cnt}; \rfloor$, користувач однозначно вважатиметься шахраєм, інакше – органічним, згідно до цього показника. Коефіцієнти $K_{5\_min\_cnt}$ та $K_{5\_max\_cnt}$ визначаються на основі експертного оцінювання (Додаток В).
k_6 – Частота кожного типу події користувача (час між подіями кожного типу)	Даний показник буде порівнюватись між користувачами, які зареєстровані у додатку однакову кількість тижнів. Шахраї зазвичай мають шаблон подій.

Продовження таблиці 2.1

Тип вхідних даних по кожному з користувачів окремо	**Формалізація вхідних даних**
Соціальні мережі: k_7 – Користувач підв'язаний до соціальної мережі чи ні k_8 – Кількість друзів у соціальній мережі (якщо підв'язаний до соціальної мережі)	Даний показник присутній не у всіх користувачів, оскільки не всі органічні користувачі мають соціальні мережі та не всі хочуть прив'язувати до них додаток.
	Коефіцієнт k_8 визначатиметься як $\frac{K_{8\_opt}}{friends\_cnt}$, де *friends_cnt* – кількість друзів користувача, K_{8_opt} – кількість друзів, при якій користувач вважається органічним. При $k_8 \in \lfloor K_{8\_min}; K_{8\_max} \rfloor$ – користувач вважається шахраєм. Інакше – користувач може бути як шахраєм, так і органічним користувачем, тому в такому випадку даним коефіцієнтом можна знехтувати. Коефіцієнти $K_{8\_opt}, K_{8\_min}, K_{8\_max}$ визначаються на основі експериментально-експертного оцінювання (Додаток В) та при порівнянні з узагальненим шаблоном шахрая аналогічно до k_2.
k_{19} – тип події – доступна / недоступна	Якісні дані, в яких, на відміну від ознаки, якій відповідає коефіцієнт k_{18}, уся множина типів подій завчасно відома. Проте, при додаванні нових типів подій та використовуючи існуючі методи подолання різнорідності, необхідно перенавчати усю систему, попередньо додавши нові категорії. Існує декілька властивостей множини вхідних даних за даною ознакою. Так, наприклад, якщо серед здійснених користувачем дій присутні недоступні для нього типи подій (так, наприклад, деякі функції мобільного додатку можуть бути доступні лише після реєстрації), то коефіцієнт приймає значення 0.

Продовження таблиці 2.1

Тип вхідних даних по кожному з користувачів окремо	**Формалізація вхідних даних**
k_9 – Кількість акаунтів з одного пристрою	Наприклад, такий додаток як Instagram [100] дозволяє мати не більше, ніж 5 акаунтів на одному пристрої, у іншому випадку він вважає пристрій шахрайським. Отже, визначатимемо даний коефіцієнт аналогічно до того, як це визначає Instagram. Якщо C_{ac} – це кількість акаунтів користувача з одного пристрою, то матимемо: $\begin{cases} k_9 = 0, при\ C_{ac} \in [0; K_{9\_min}] \\ k_9 = 1, при\ C_{ac} \in (K_{9\_min}; \infty) \end{cases}$ Зазначимо, що k_9 = 0 означає, що користувач є шахраєм. k_9 = 1 – користувач є органічним. Коефіцієнт K_{9_min} визначається на основі експериментально-експертного оцінювання (Додаток В) та при порівнянні з узагальненим шаблоном шахрая аналогічно до k_2.
k_{10} – Кількість інсталювань з одного пристрою	Тому даний коефіцієнт визначатимемо за наступною формулою: $k_{10} = 1 - \frac{1}{k_{times}}$, k_{times} – кількість інсталювань з пристрою користувача (мінімальна кількість інсталювань рівна 1). Якщо користувач зробив одне інсталювання, то даний коефіцієнт k_{10} буде рівний 0, і при більшій кількості інсталювань, тим даний коефіцієнт буде більше наближатись до 1. Проте, даний коефіцієнт повинен працювати в парі з наступним – k_{11}.

Продовження таблиці 2.1

Тип вхідних даних по кожному з користувачів окремо	**Формалізація вхідних даних**
k_{11} – Час між інсталюваннями на одному пристрої	При дуже частих інсталюваннях з одного пристрою раціонально запідозрити шахрайську діяльність, яка проводиться з цього пристрою. Якщо частота інсталювання з одного пристрою, яка задається коефіцієнтом k_{11}, показує, що час між запитами на інсталювання не менший, ніж k_{11_min} то за даними коефіцієнтами користувач вважатиметься органічним. Проте, при визначенні даного коефіцієнта необхідно перевірити усі інші показники. Коефіцієнт k_{11_min} визначається на основі експертного опитування та при порівнянні з узагальненим шаблоном шахрая аналогічно до k_2.
k_{12} – Верифікація IP: входить до списку шахрайських чи ні	Якщо вже існує список шахрайських IP-адрес (уся множина користувачів з цих IP-адрес були шахраями), то можна вважати, що всі користувачі з даною IP-адресою є шахраями. Отже, коефіцієнт у цьому випадку завідома буде $k_{12}=0$, інакше – $k_{12}=1$.
k_{13} – Час інсталювання	Як було наведено у пункті 1.1.3, час інсталювання повинен входити у заданий проміжок, який задається мінімальним та максимально допустимим значеннями проміжку часу. Проміжок часу T_i визначається експериментально відомими органічними користувачами та залежить від їх інтернету (мобільний чи Wi-Fi, швидкість інтернету), доступних параметрів пристрою (пам'ять тощо). Для встановлення граничного мінімального значення часу інсталювання T_{i_min} проводиться експеримент з Wi-Fi, з нового пристрою. Якщо ж у користувача час інсталюапння $T_{install} < T_{i\_min}$, то $k_{13}=0$, що означає, що користувач є шахраєм.
k_{18} – ідентифікатор пристрою користувача	Визначається аналогічно коефіцієнту k_{12}.

Продовження таблиці 2.1

Тип вхідних даних по кожному з користувачів окремо	**Формалізація вхідних даних**
k_{14} – Інформація про користувача	Даний коефіцієнт k_{14} рівний максимальному значенню коефіцієнта схожості по інформації між даним користувачем та іншими існуючими. Простіше кажуче, визначає, чи даний користувач не використовує у профілі інформацію інших користувачів. Так, наприклад, Instagram визначає таких користувачів шахраями. Формалізуємо визначення даного коефіцієнта для користувача $user_x$ з використанням теорії множин. Якщо існує інформація по даному користувачу u_x та існує частково впорядкована множина інформації по іншим користувачам мобільного додатку $U = \{u_1, u_2, ..., u_n\}$, які зареєструвалися раніше даного користувача u_x, то існує частково впорядкована множина $K = \{k_{x,1}, k_{x,2}, ..., k_{x,n}\}$, яка задає коефіцієнт схожості інформації користувача u_i з інформацією даного користувача u_x. Зазначимо, що інформація по користувачам задається масивом слів. Коефіцієнт $k_{x,i}$ визначає, який відсоток інформації користувача u_x схожий до інформації користувача u_i. Тобто $k_{x,i} = \frac{ux}{ui}$. У свою чергу, конфіцієнт k_{14} є максимальним елементом частково впорядкованої множини K, тобто $k_{14} = m$, де $m \in K$ та $\forall k \in K : (m \le k \Rightarrow k = m)$. Якщо інформація даного користувача співпадає з інформацією будь-якого раніше зареєстрованого користувача хоча б на k_{14_opt}% (тобто $k_{14} \in \left[\frac{k_{14\_opt}}{100}; 1\right]$), то даного користувача вважатимемо шахраєм. Коефіцієнт k_{14_opt} визначається на основі експериментально-експертного оцінювання (Додаток В). До речі, таке ж правило працює при перевірці дисертаційних та дипломних робіт на плагіат.

Продовження таблиці 2.1

Тип вхідних даних по кожному з користувачів окремо	**Формалізація вхідних даних**
k_{15} – Фото користувача	Аналогічно до попереднього коефіцієнту визначається і коефіцієнт k_{15}, який визначає, якщо фото даного користувача повністю співпадає з фото попередньо зареєстрованого користувача і $k_{15} \in \left[0; \frac{k_{15\_min}}{100}\right]$ (різниці між пікселями фото немає взагалі або ж фото відрізняється до k_{15_min}%), то даний користувач вважатиметься шахраєм. Так, наприклад, можна використати модуль створення шахрайських Instagram-акаунтів SocialKit [101], який пропонує заміняти 1% пікселей у чужому фото для того, щоб алгоритм Instagram не зміг визначити, що фото є шахрайським. Ми ж для точності будемо вважати шахраями тих, у кого фото має різницю у $0 - k_{15\_min}$ % пікселей. При значенні коефіцієнта $k_{15} \in (k_{15\_min}; 1]$ користувач не є шахраєм. Коефіцієнт k_{15_min} визначається на основі експериментально-експертного оцінювання (Додаток В).
k_{17} – Кількість кліків з однієї IP-адреси за хвилину	Визначається аналогічно до k_1, лише базується на данних з IP-адреси: $k_{17} = \frac{\sum_1^y ri}{Y_{avg}}$, де $X_{17} = \{r_1, r_2, ...r_y\}$ – вектор кількості кліків з IP-адреси даного користувача по хвилинам, $Y_{avg} = \frac{Y_{1avg} + Y_{2avg} + ... + Y_{Ravg}}{R}$, де R – кількість відомих IP-адрес, які використовуються органічними користувачами, $Y_{1avg}, Y_{2avg}, ...Y_{Ravg}$ – це середня кількість кліків з IP-адрес, якими користуються органічні користувачі у першу, другу, …, R-ту хвилини відповідно.

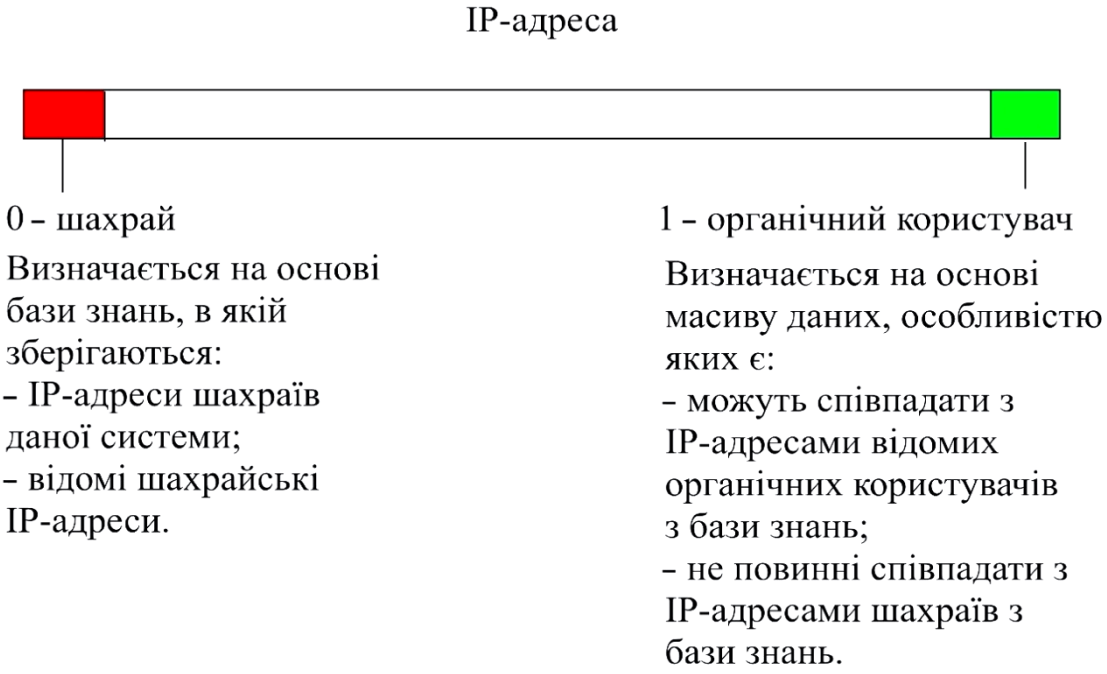


Рис. 2.15. Шкала визначення класу користувача за ознакою «IP-адреса»

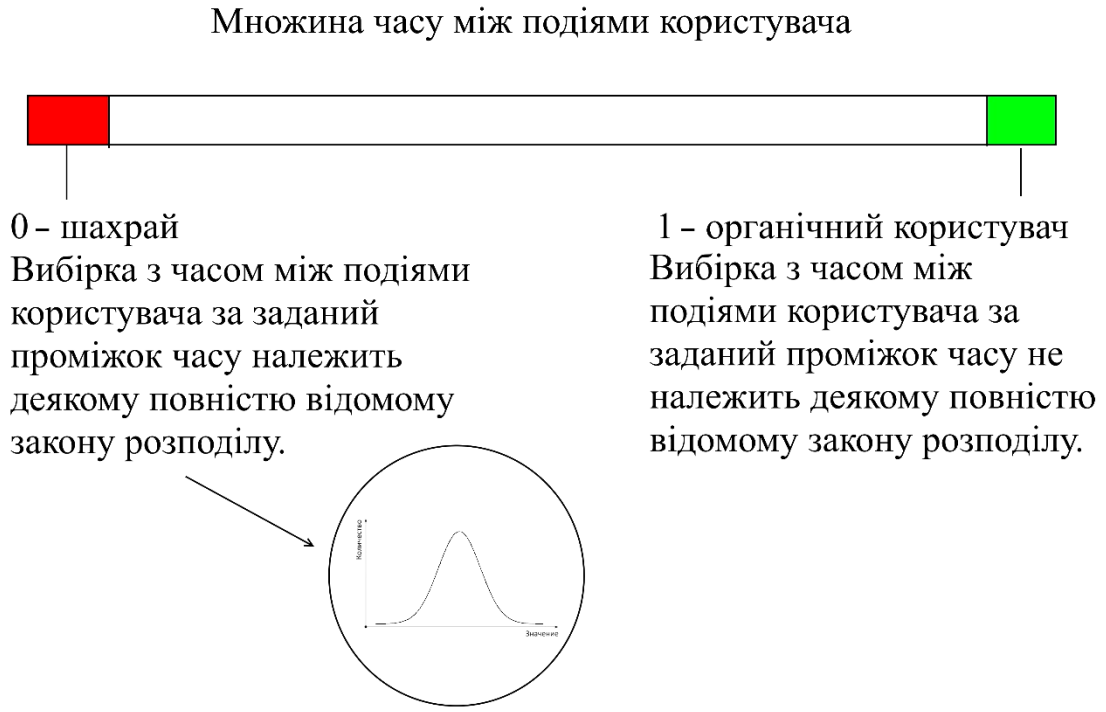


Рис. 2.16. Шкала визначення класу користувача за ознакою «множина часу між подіями користувача»

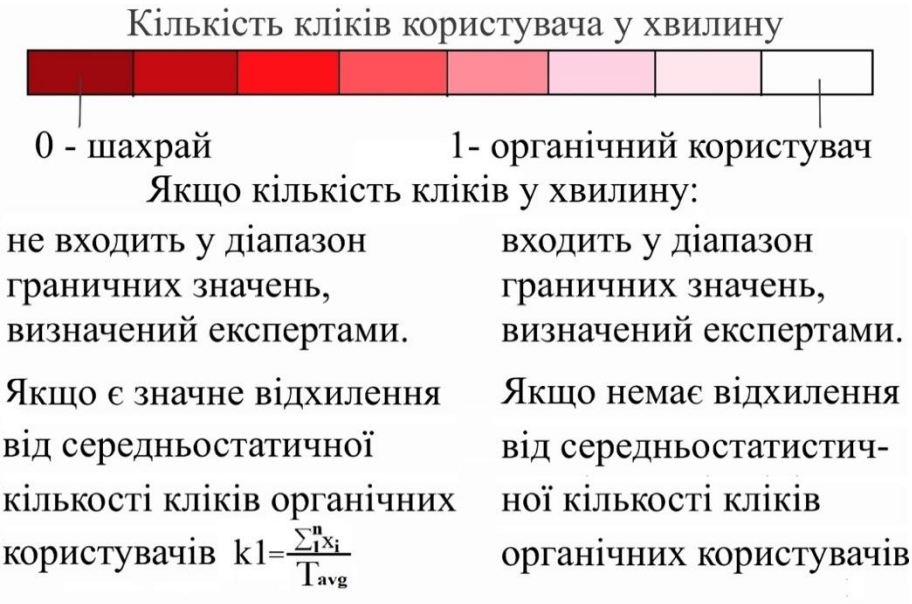


Рис. 2.17. Шкала визначення класу користувача за ознакою «кількість кліків користувача за хвилину»

- k_2 – куплена / не куплена покупка;

- k_3 – підтверджена / не підтверджена покупка;

- k_4 – час між подіями користувача;

- k_5, k_6 – тип подій користувача, частота кожного типу події користувача – суміжні між собою показники;

- k_{12} – верифікація IP – входить до списку шахрайських чи ні;

б) **друга група** охоплює коефіцієнти, за якими неможливо зробити первинний аналіз. Проте слід зазначити, що на основі коефіцієнтів першої групи було сформовано бази знань і бази даних однозначних шахрайських і органічних користувачів. Маючи ці дані, було визначено коефіцієнти схожості по кожній характеристиці другої групи між користувачами з сформованої бази даних та невизначеними користувачами. Так, до другої групи відноситься 6 коефіцієнтів (деякі з них співпадають з першою групою). Деякі з них:

- k_1 (кількість кліків з одного пристрою за хвилину) – важливий коефіцієнт при виявленні шахрайства, проте маючи лише його, не можна однозначно виявити шахрая (аномалію);

- k_4 (час між подіями користувача) – аналогічно до коефіцієнту k_1, якщо дії не підпадають певному шаблону та час між подіями не заданий розподілом;

- k_8 (кількість друзів у соціальній мережі) – якщо користувач прив'язаний до соціальної мережі та в нього є достатня кількість друзів, а саме – $\left(k_{8\_max} * k_{8\_opt}; \infty\right)$, то можна зробити перевірку на імена (чи друзі реальні) і фото друзів (аналогічно до визначення коефіцієнта k_{15});

- k_{10}, k_{11} – кількість інсталювань з одного пристрою, час між інсталюваннями на одному пристрої – дозволяє визначити, чи користувач є органічним на основі даних про інсталювання. Проте при визначенні даного коефіцієнта необхідно перевірити усі інші показники.

2.3.1.2 Розробка моделей процесу подолання різнорідності вхідних даних

Проведений вище аналіз даних, а також досвід розробників систем-аналогів показав, що разом всі ці дані неможливо використовувати і, крім того, що вони різнорідні, їх дуже багато. І хоча сучасні методи інтелектуального аналізу даних можуть аналізувати величезну кількість даних, вони не можуть подолати їх різнорідність (рис. 2.11).

Таким чином, при шкалюванні вхідних даних, відповідно запропонованих в роботі шкал (рис. 2.15–2.17), отримано математичну модель процесу шкалювання (2.29), яка містить коефіцієнти визначеної метрики.

$$\vec{I}\begin{pmatrix} U_1(i_1,i_2,...,i_{s1}) \\ U_2(i_1,i_2,...,i_{s2}) \\ ... \\ U_n(i_1,i_2,...,i_{sn}) \end{pmatrix} \rightarrow$$

$$\rightarrow \begin{cases} \vec{G}_1\begin{pmatrix} U_1(g_{11},...,g_{1r}) \\ U_2(g_{11},...,g_{1r}) \\ ... \\ U_n(g_{11},...,g_{1r}) \end{pmatrix} \rightarrow \begin{cases} \vec{X}\begin{pmatrix} U_1(x_1,...,x_n) \\ U_2(x_1,...,x_n) \\ ... \\ U_n(x_1,...,x_n) \end{pmatrix} \rightarrow F_1(\vec{X}) \rightarrow \vec{X}_1\begin{pmatrix} U_n(k_{01}) \\ U_n(k_{02}) \\ ... \\ U_n(k_{0n}) \end{pmatrix} \\ \vec{B}\begin{pmatrix} U_1(b_1,...,b_k) \\ U_2(b_1,...,b_k) \\ ... \\ U_n(b_1,...,b_k) \end{pmatrix} \rightarrow F_2(\vec{B}) \rightarrow \vec{B}_1\begin{pmatrix} U_n(k_{11}) \\ U_n(k_{12}) \\ ... \\ U_n(k_{1n}) \end{pmatrix} \rightarrow \\ \vec{W}\begin{pmatrix} U_1(w_1,...,w_m) \\ U_2(w_1,...,w_m) \\ ... \\ U_n(w_1,...,w_m) \end{pmatrix} \rightarrow F_3(\vec{W}) \rightarrow \vec{W}_1\begin{pmatrix} U_n(k_{21}) \\ U_n(k_{22}) \\ ... \\ U_n(k_{2n}) \end{pmatrix} \end{cases} \\ \vec{G}_2\begin{pmatrix} U_1(g_{21},...,g_{2l}) \\ U_2(g_{21},...,g_{2l}) \\ ... \\ U_n(g_{21},...,g_{2l}) \end{pmatrix} \rightarrow \end{cases}$$

$$\rightarrow F_4(\vec{X}_1,\vec{B}_1,\vec{W}_1) \rightarrow \vec{D}\begin{pmatrix} U_1(k_{01},k_{11},...,k_{21}) \\ U_2(k_{02},k_{12},...,k_{22}) \\ ... \\ U_n(k_{0n},k_{1n},...,k_{2n}) \end{pmatrix} \rightarrow F_5(\vec{D}) \rightarrow \vec{R}\begin{pmatrix} U_1(C_1) \\ U_2(C_0) \\ ... \\ U_n(C_1) \end{pmatrix}, \quad (2.29)$$

де $\vec{G}_1$ та $\vec{G}_2$ – вектори різнорідних вхідних даних, поділені на дві групи, а саме так, як визначено на основі експертного опитування;

$\vec{X},\vec{B},...,\vec{W}$ – вектори однорідних даних, поділених по типам;

$\vec{I}\begin{pmatrix} U_1(i_1,i_2,...,i_{s1}) \\ U_2(i_1,i_2,...,i_{s2}) \\ ... \\ U_n(i_1,i_2,...,i_{sn}) \end{pmatrix}$ – інформація з бази даних по кожному з користувачів, а саме вектор, елементами якого є множини усіх визначених ознак по кожному з користувачів $(U_1,U_2,...,U_n)$;

$F_1(\vec{X}), F_2(\vec{B}), ..., F_3(\vec{W})$ – відповідні функції переводу однорідних даних за певною ознакою у критерій, значення якого від 0 до 1. На виході буде отримано вектори $\vec{X}_1, \vec{B}_1, \vec{W}_1$, що міститимуть користувачів із значенням критерію по відповідній ознаці;

$F_4(\vec{X}_1, \vec{B}_1, ..., \vec{W}_1)$ – функція об'єднання усіх критеріїв по користувачам у вектор $\vec{D}$;

$\vec{D}$ – вектор уніфікованих ознак без зменшення діагностичної цінності інформації;

$F_5(\vec{D})$ – функція класифікації користувачів на кластери C_0 (шахраї) та C_1 (органічні користувачі).

Результат класифікації буде представлено у вигляді вектору користувачів $\vec{R}$, кожен з користувачів якого міститиме у якості параметру відповідний клас.

На основі запропонованої математичної моделі шкалювання (2.29) розроблено три алгоритми процесу подолання різнорідності вхідних даних. Розглянемо ці алгоритми більш детально.

На думку автора, для того, щоб використовувати всі наявні різнорідні дані, необхідно провести їх шкалювання, так як це є одним із стандартних підходів подолання різнорідності. Але у зв'язку із складністю та кількістю даних, здійснимо шкалювання не однією, а декількома шкалами, об'єднуючи дані по групам. Такий підхід був запропонований автором роботи у вигляді трьох алгоритмів процесу подолання різнорідності вхідних даних на базі шести шкал згрупованих даних.

Алгоритм 1 подолання різнорідності вхідних даних при інсталюванні мобільних додатків:

1. Отримання даних $\bar{I}\begin{pmatrix} U_1(i_1, i_2, ..., i_{s1}) \\ U_2(i_1, i_2, ..., i_{s2}) \\ ... \\ U_n(i_1, i_2, ..., i_{sn}) \end{pmatrix}$.

2. Визначення типу даних.

3. Перетворення даних, в залежності від їх типу, алгоритмами 2 та 3 з метою зведення їх до однорідних даних.

Для виконання пункту 3 алгоритму 1, пропонуємо алгоритм 2 шкалювання даних, який дає можливість переходу від різнорідних даних до бінарних, значення яких можуть бути або 0, або 1, за допомогою запропонованих в роботі коефіцієнтів. Де 0 означатиме, що користувач є шахраєм, а 1 позначатиме користувача як органічного. Бінарність даних після перетворення визначається кінцевою метою задачі – чи користувач з певними даними є шахраєм чи ні. Тому в першу групу G_1 входять дані, результатом аналізу яких є бінарна відповідь «так» або «ні» (0 або 1). А ті дані, які неможливо привести до бінарного типу, групуються у другу групу даних G_2, для яких у даній роботі розроблений метод (на базі алгоритму 3) інтелектуального аналізу даних з використанням коефіцієнтів схожості та системи правил з сформованою базою знань. Якщо ж даний метод не визначає користувача ні як шахрая, ні як органічного, то даний користувач включається в групу підозрілих користувачів, для аналізу яких застосовуються алгоритми нечіткої логіки на основі попередньо сформованих правил з бази знань, яка в процесі експлуатації постійно нарощується.

Слід зазначити, що не всі вхідні дані можна відразу однозначно шкалювати. Тому в роботі розроблений алгоритм 2 шкалювання вхідних даних, який оснований на поділі всіх даних на 2 групи, над однією з яких однозначно проводиться шкалювання. Представимо алгоритм 2 шкалювання вхідних даних:

1. Поділ усіх даних $\vec{I}$ на дві групи G_1 і G_2.

1.1. До першої групи G_1 входитимуть дані, за якими однозначно можна буде визначити значення 0 або 1 за відповідним коефіцієнтом, де 0 визначатиме користувача як шахрая, а 1 визначатиме користувача органічним (шкала 1, 2, 3 рис. 2.18). Так, наприклад, значення коефіцієнту k_{16}, що відповідає визначенню шахрайства за IP-адресою, визначатиметься за допомогою перевірки належності IP-адреси користувача IP до множини відомих шахрайських IP-адрес $FRAUD\_IP$, а саме, якщо виконується умова $P_1(IP) = IP \in FRAUD\_IP$, то

користувач є шахраєм. Аналогічно, наприклад, значення коефіцієнту k_3, що відповідає визначенню шахрайства за унікальним ідентифікатором пристрою користувача $deviceID$, визначатиметься умовою $P_3(deviceID) = deviceID \in FRAUD\_deviceID$, виконання якої помітить користувача шахраєм. Зазначимо, що $FRAUD\_deviceID$ – це множина усіх відомих ідентифікаторів пристроїв шахраїв. Значення по коефіцієнту k_5 буде рівне 0, якщо користувач відправив запит з ідентифікатором покупки $purchaseID$, який не був підтверджений мобільним магазином (атрибут покупки $isConfirmed$ позначений як *false*). У цьому випадку визначення шахрая можна задати умовою $P_5(purchaseID) = (purchaseID.isConfirmed = false)$.

1.2. До другої групи G_2 входитимуть дані, за якими неможливо однозначно визначити значення коефіцієнту. Так, наприклад, невідомі граничні значення часу між подіями, за допомогою яких можна однозначно визначити чи користувач є шахраєм, чи органічним користувачем. Один із коефіцієнтів визначатиметься на основі типів подій, які робить користувач. Даний коефіцієнт не може входити до першої групи G_1, оскільки по таким даним не існує чітко визначеної умови, яка не буде змінюватися з часом.

2. Визначення значень коефіцієнтів на основі даних першої групи G_1 та побудова моделі даних групи G_1 (рис. 2.18).

3. Визначення однозначних шахраїв, органічних та підозрілих користувачів на основі значень коефіцієнтів першої групи G_1, формування їх шаблонів та занесення їх у базу знань.

Результатом шкалювання за першою групою представлена таблиця, фрагмент якої подано у таблиці 2.2 [16]. На основі таких таблиць створюється база даних шахраїв і тому перший етап – це визначення відомих шахраїв по цим таблицям.

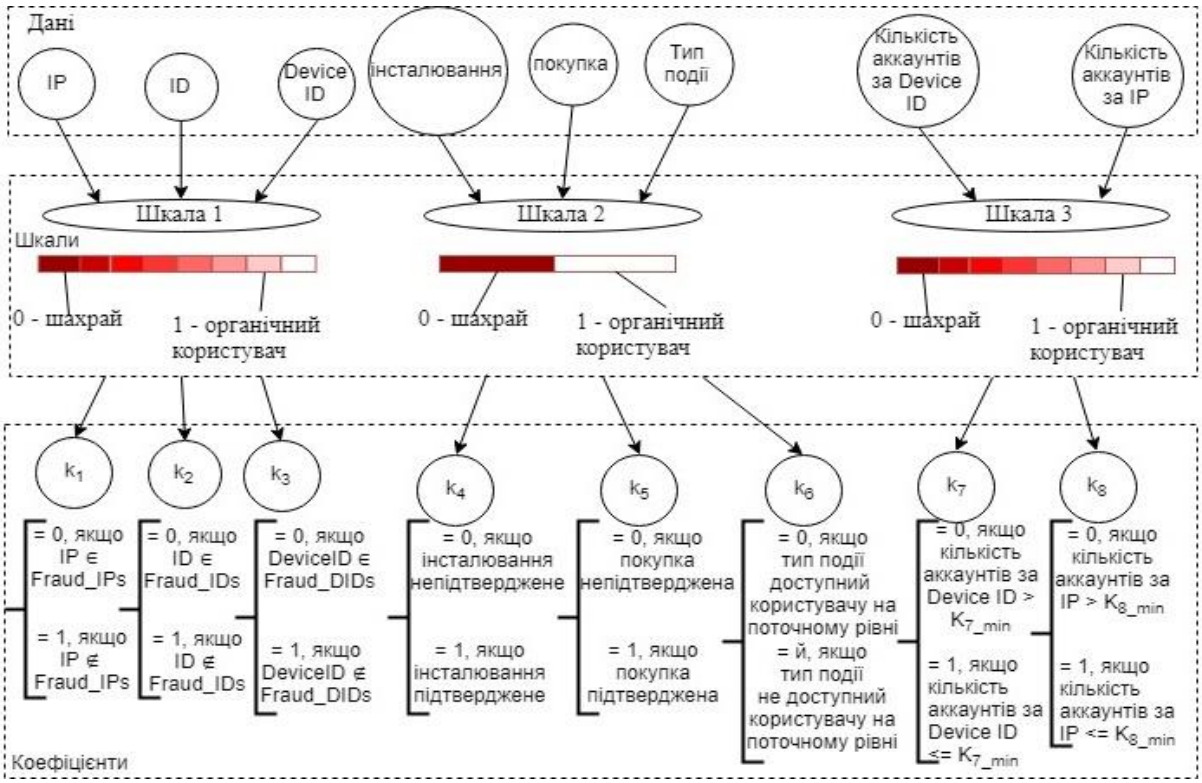


Рисунок 2.18 – Модель даних першої групи G_1

Таблиця 2.2

Приклад принципу роботи запропонованого методу подолання різнорідності вхідних даних

Ознака	**Коефіцієнт**		**Умова**	**Розпізнавання шахрайства**
	До шкалювання	Після шкалювання		
IP-адреса	127.0.0.1	1	$IP \notin FRAUD\_deviceIP$	органічний користувач
ID пристрою	355776785678735	0	$deviceID \in FRAUD\_deviceID$	шахрай
ID покупки підтверджена чи ні	6cf9094a-6fbf-4898-bb3e-0cd895b1cafb	0	$purchaseID.isConfirmed = false$	шахрай

4. Визначення характеристик однозначно визначених користувачів, формування їх шаблонів та занесення їх у базу знань.

Для аналізу другої групи даних G_2 в роботі розроблено метод на основі інтелектуального аналізу даних, оснований на алгоритмі 3 процесу подолання різнорідності вхідних даних:

1. Визначення початкового узагальненого портрету шахрая.

2. Визначення значень коефіцієнтів другої групи даних G_2 на основі правил попередньо сформованої бази знань (шкали 4, 5 рис. 2.19) [13]:

2.1. Отримання шаблонів та характеристик шахраїв з бази даних, що була сформована у процесі аналізу групи G_1.

2.2. Знаходження коефіцієнту схожості між поточним користувачем та шаблоном і характеристиками шахрая (рис. 2.19). Зазначимо, що значення коефіцієнтів схожості належить проміжку [0; 1]. Значення 1 означає, що користувачі ідентичні за даною ознакою, 0 означає, що користувачі не мають нічого спільного за даною ознакою. Так, наприклад, розглянувши такі дані як час між подіями, матимемо множину часу між подіями поточного користувача $T_u = \{t \mid t \geq 0\}$ та множини часу між подіями кожного із однозначно визначених на етапах алгоритму 3, 4 шахраїв $T_i = \{t \mid t > 0\}$. Маючи дві множини не бінарних, проте однорідних даних T_u та T_i, застосуємо відповідний коефіцієнт схожості користувачів. Для множин такого типу в роботі обрано коефіцієнт Танімото $K_T = (T_u, T_i)$ [2], [19], [63], який визначається як $K_T(T_u, T_i) = \frac{N_c}{N_a + N_b - N_c}$, де N_c – кількість спільних для множин T_u та T_i елементів, N_a – кількість елементів у множині T_u, N_b – кількість елементів у множині T_i. У свою чергу, для множин бінарних даних, таких як множина з бінарними значеннями по кожному з існуючих типів подій мобільного додатку, де 0 означатиме, що користувач не використовував таку подію, а 1 означатиме, що використовував, коефіцієнти схожості користувачів визначаються з використанням коефіцієнту косинусної схожості $K_{cos}(A_1, A_2)$ [2], [19], [63]. Зазначимо, що $K_{cos}(A_1, A_2) = cos(A_1, A_2) = \frac{(A_1 * A_2)}{|A_1| * |A_2|}$, де A_1, A_2 – множини з вище вказаними бінарними даними поточного користувача та шахрайського користувача відповідно.

2.3. Формування масиву значень коефіцієнтів схожості поточного користувача з кожним із однозначно визначених шахрайських користувачів G_{2_fraud} з бази знань.

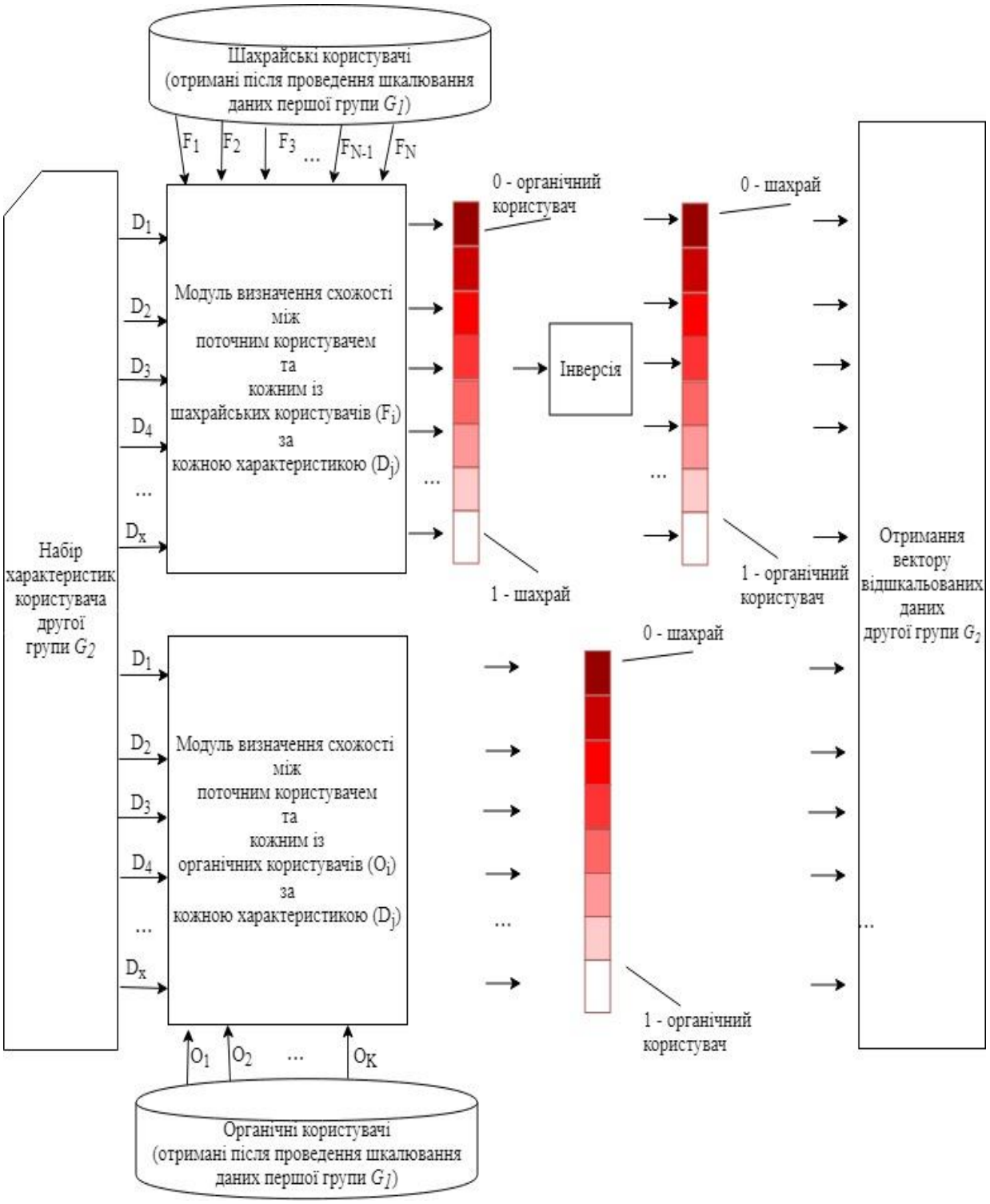


Рисунок 2.19 – Модель шкалювання даних другої групи G_2

2.4. Виконання кроків 2.1 – 2.3 для однозначно визначених органічних користувачів G_{2_org} з бази знань.

2.5. Інверсія масиву значень коефіцієнтів схожості поточного користувача з кожним із однозначно визначених органічних користувачів G_{2_org} з бази знань.

2.6. Формування спільного масиву коефіцієнтів $G_2 = G_{2\_fraud} \cup \left(1 - G_{2\_org}\right)$ по поточній ознаці на основі масивів, отриманих на кроках 2.3, 2.4, 2.5.

2.7. Формування вектору підозрілих користувачів G_{2_susp}.

3. Формування вектору відшкальованих даних по кожному користувачу за коефіцієнтами, що об'єднує результати, отримані на кроці 2 алгоритму 2 та кроці 2.7 алгоритму 3. Сформований вектор є вектором уніфікованих ознак без зменшення діагностичної цінності інформації.

4. Віднесення підозрілих користувачів до класу шахраїв або органічних з використанням нечіткої логіки.

В результаті виконання кроків 1 та 2 алгоритму 2, проведено шкалювання даних групи G_1 та за допомогою коефіцієнтів k_1, k_2, k_3, …, k_8 різнорідні дані приведено до однорідного стану, що дозволяє однозначно визначити шахраїв, якщо значення хоча б одного з коефіцієнтів дорівнює 0.

Зазначимо, що запропоновані метод та алгоритми можна автоматизувати. Також, з кожним повторним використанням алгоритмів, будуть знаходитись все нові характеристики органічних та шахрайських користувачів, які заноситимуться у базу знань. Це дасть змогу удосконалювати подальше виявлення шахрайства.

2.3.2 Аналіз доцільності використання методів обробки Big Data для виявлення аномалій в даних при інсталяції мобільних додатків

Для виявлення шахрайства як аномалій в даних при інсталюванні мобільних додатків доцільно використовувати методи обробки великих даних (Big Data).

Перед початком аналізу необхідно визначити поняття великих даних (Big Data). Як зазначено у [89], Big data – це різні інструменти, підходи і методи обробки як структурованих, так і неструктурованих даних для того, щоб їх використовувати для конкретних завдань і цілей. У даному контексті неструктуровані дані – це інформація, яка не має заздалегідь певної структури або не організована в певному порядку. Для великих даних виділяють традиційні визначальні характеристики, вироблені Meta Group ще в 2001 році, які називаються «Три V»:

– об’єм (volume) – величина фізичного обсягу;

– швидкість (velocity) – швидкість приросту і необхідності швидкої обробки даних для отримання результатів;

– різноманітність (variety) – можливість одночасної обробки різних типів даних.

З вищезазначеного можна відзначити, що у поставленій задачі виявлення шахрайства як аномалії в даних при інсталюванні мобільних додатків дані про користувачів мають усі властивості великих даних. Тому використання методів обробки великих даних для виявлення аномалій в даних при інсталяції мобільних додатків є доцільним.

Одним із таких методів є класифікація вхідних даних, тому розглянемо та розробимо модель класифікації вхідних даних для виявлення аномалій в Big Data.

2.3.3 Модель класифікації вхідних даних для виявлення аномалій в Big Data

Оскільки, як було зазначено у пп. 2.3.1.1, в процесі шкалювання та на основі результатів експертного оцінювання в роботі було розроблено 17 шкал. Для конвертації наявних даних до значень на розроблених шкалах, у роботі було сформовано 17 коефіцієнтів, які впливають на прийняття рішень при виявленні шахрайства. Аналіз цих коефіцієнтів дозволив здійснити поділ коефіцієнтів на

дві групи. Перша група охоплює коефіцієнти, які дозволяють провести попередній аналіз даних, а саме, однозначно з множини користувачів визначити шахрайських користувачів, органічних та підозрілих. А друга група охоплює коефіцієнти, за якими неможливо зробити первинний аналіз. Саме через наявність другої групи коефіцієнтів виникла необхідність у побудові моделі класифікації, яка б на основі отриманої множини однорідних значень всіх наявних даних з використанням алгоритму 1 подолання різнорідності вхідних даних при інсталюванні мобільних додатків та алгоритму 2 шкалювання вхідних даних, дозволяла б визначити клас користувача. Це дало б змогу в подальшому автоматизувати процес виявлення шахрайства, створити портрет шахраїв, що містив би навіть нові та непомітні експертам характеристики шахрая.

Отже, отримавши множину однорідних значень всіх наявних даних за допомогою запропонованих шкал, виконаємо етап класифікації даних з використанням відомої моделі класифікації (екстремальне градієнтне підсилення XGBoost, випадковий ліс – random forest, глибокі нейронні мережі) з метою виявлення даних, які однозначно визначають шахраїв, і даних, які однозначно визначають органічних користувачів.

У даній роботі у якості моделі класифікації було обрано повністю-зв’язану глибоку нейронну мережу (fully-connected deep neural network – DNN) з трьома прихованими шарами, оскільки штучні глибокі нейронні мережі дозволяють розв’язувати задачі розпізнавання і класифікації з високою точністю.

Удосконалену модель класифікації вхідних даних для виявлення аномалій в Big Data на основі глибинної нейронної мережі з трьома прихованими шарами, що використовується у даній роботі, представлено на рис. 2.20.

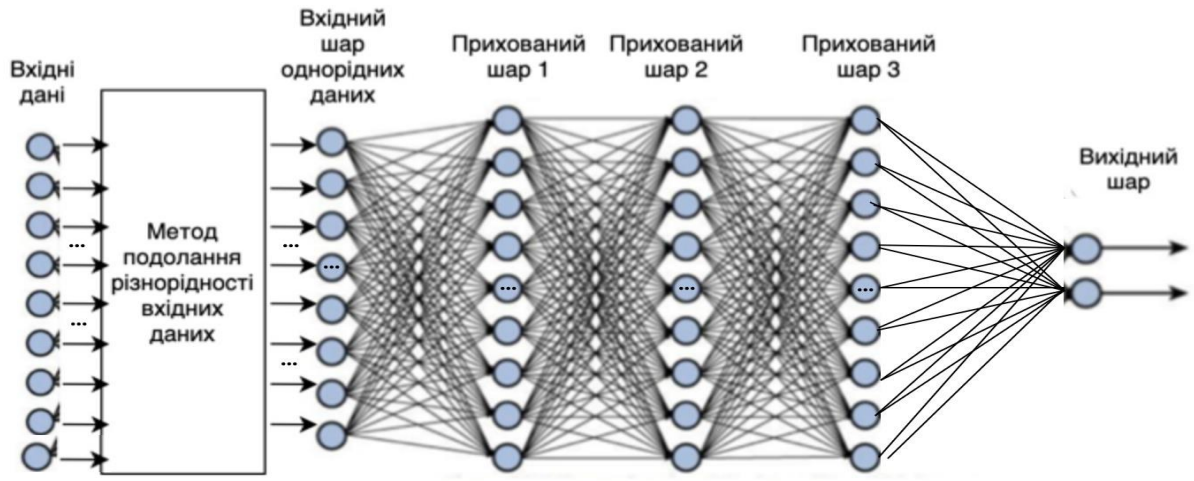


Рисунок 2.20 – Модель класифікації вхідних даних для виявлення аномалій в Big Data

Узагальнену математичну модель процесу класифікації представлено у (2.27).

$$\vec{I}\begin{pmatrix} U_1(i_1,i_2,...,i_{s1}) \\ U_2(i_1,i_2,...,i_{s2}) \\ ... \\ U_n(i_1,i_2,...,i_{sn}) \end{pmatrix} \to P(\vec{I}) \to \vec{D}\begin{pmatrix} U_1(k_{01},k_{11},...,k_{21}) \\ U_2(k_{02},k_{12},...,k_{22}) \\ ... \\ U_n(k_{0n},k_{1n},...,k_{2n}) \end{pmatrix} \to$$

$$\begin{matrix} \to K(\vec{I},\vec{D}) \\ \to S(\vec{D}) \\ \to B(\vec{I},\vec{D}) \end{matrix} \to \vec{X}\begin{pmatrix} X_{train} \\ Y_{train} \\ X_{test} \\ Y_{test} \end{pmatrix} \to C_{build}(\vec{X}) \to M \to$$

$$\to C_{check}(M, X_{val}, Y_{val}) \to M_{checked} \to \vec{R}\begin{pmatrix} U_1(C_1) \\ U_2(C_0) \\ ... \\ U_n(C_1) \end{pmatrix}, \qquad (2.27)$$

де $\vec{I}\begin{pmatrix} U_1(i_1,i_2,...,i_{s1}) \\ U_2(i_1,i_2,...,i_{s2}) \\ ... \\ U_n(i_1,i_2,...,i_{sn}) \end{pmatrix}$ – інформація з бази даних по кожному з користувачів, а

саме вектор, елементами якого є множини усіх визначених ознак по кожному з користувачів $(U_1,U_2,...,U_n)$;

$P(\vec{I})$ – розроблений в роботі узагальнений метод подолання різнорідності вхідних даних, що на вхід приймає вектор $\vec{I}$;

$\vec{D}$ – вектор з усіма однорідними та нормалізованими критеріями по усім користувачам, а саме – вектор уніфікованих ознак без зменшення діагностичної цінності інформації;

$B(\vec{I},\vec{D})$ – функція побудови бази даних користувачів;

$K(\vec{I},\vec{D})$ – функція побудови бази знань;

$S(\vec{D})$ – функція поділу вектору однорідних помічених даних на тренувальну та тестувальну вибірку, що на виході дає вектор $\vec{X}$;

X_{train} – набір вхідних даних про користувачів з тренувальної вибірки;

Y_{train} – набір вихідних даних про користувачів з тренувальної вибірки;

X_{test} – набір вхідних даних про користувачів з тестувальної вибірки;

Y_{test} – набір вихідних даних про користувачів з тестувальної вибірки;

$C_{build}(\vec{X})$ – функція побудови (тренування, тестування) моделі класифікації;

M – модель класифікації;

C_{check} – функція валідації моделі класифікації;

X_{val} – набір вхідних даних про користувачів з валідаційної (контрольної) вибірки;

Y_{val} – набір вихідних даних про користувачів з валідаційної (контрольної) вибірки;

$M_{checked}$ – перевірена кінцева модель класифікації;

$\vec{R}$ – вихідний вектор користувачів поділених на кластери C_0 (шахраї) та C_1 (органічні користувачі).

У даній роботі удосконалено модель класифікації за рахунок використання розробленого методу подолання різнорідності вхідних даних замість стандартних методів перетворення вхідних даних, що дозволило підвищити точність виявлення шахрайства.

2.4 Висновки до розділу 2

Формалізовано процес виявлення шахрайства як аномалії в даних з використанням теорії множин, що дозволило визначити властивості аномальних і неаномальних даних. Це дало змогу визначити властивості даних, які позначають шахраїв, у визначеній предметній області та спростити процес пошуку шахрайства при інсталюванні мобільних додатків.

Здійснено класифікацію різнорідних даних при інсталюванні мобільних додатків на основі процесу вибору ознак, що дозволило спростити процес аналізу різнорідних за метриками, розмірностями і шаблонами даних та автоматизувати його.

Розроблено узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків, відмінність якого полягає у використанні запропонованої моделі класифікації користувачів та методу подолання різнорідності вхідних даних, що дозволяє визначити класи користувачів та підвищити точність виявлення шахрайства при інсталюванні мобільних додатків.

Запропоновано метод подолання різнорідності вхідних даних, що являє собою сукупність процедур вибору ознак, зниження розмірності та нормалізації даних, відмінність якого полягає у новій моделі процесу подолання різнорідності даних шляхом шкалювання за інформативністю, що дозволяє всю множину різнорідних даних про користувачів звести до вектору уніфікованих ознак без зменшення діагностичної цінності інформації.

Удосконалено модель класифікації користувачів на основі глибинних нейронних мереж у частині зниження розмірності та нормалізації даних згідно запропонованого методу подолання різнорідності даних, яка є основою для створення узагальненого портрету шахрая з метою спрощення процесів їх виявлення.

Результати проведених досліджень опубліковані в [13], [14], [16], [17], [19]-[22], [25], [108].

РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВИЯВЛЕННЯ ШАХРАЙСТВА ПРИ ІНСТАЛЮВАННІ МОБІЛЬНИХ ДОДАТКІВ

Для виявлення шахрайства як аномалій в даних при інсталюванні мобільних додатків розв’язується декілька задач: задача виявлення характеристик даних користувача, задача подолання різнорідності даних, задача навчання моделі класифікації, задача класифікації, задача формування бази знань для виявлення шахрайства, задача формування бази даних шахраїв, задача інтелектуального аналізу наявних даних та формування портрету користувача, задача створення узагальненого портрету шахрая та прогнозування (prediction). Розв’язання кожної з цих задач вимагає визначення даних, які обробляються, процесів та засобів їх обробки. Тому для ефективного функціонування інформаційної технології виявлення шахрайства на основі розробленого в роботі узагальненого методу, який оснований на моделях, методах та алгоритмах виявлення шахрайства як аномалій в даних, необхідно розробити концептуальні основи побудови інформаційної технології.

3.1 Концептуальні особливості побудови інформаційної технології виявлення шахрайства на основі методів та моделей

На основі розглянутих задач, необхідних для розробки узагальненого методу виявлення шахрайства при інсталюванні мобільних додатків, запропонуємо інформаційну технологію виявлення шахраїв на базі інтелектуального аналізу даних, яка складатиметься з наступних блоків:

– блок виявлення характеристик даних користувача;

– блок подолання різнорідності даних;

– блок побудови моделі класифікації;

– блок класифікації;

– блок формування бази знань для виявлення шахрайства;

– блок формування бази даних шахраїв;

– блок інтелектуального аналізу наявних даних та формування портрету користувача;

– блок створення узагальненого портрету шахрая та прогнозування.

При побудові інформаційної технології наявність шахраїв в роботі розглядається як наявність аномалії в великих масивах різнорідних даних про користувачів.

3.1.1 Розробка концептуальної моделі інформаційної технології виявлення шахрайства

При проектуванні основних компонентів інформаційної технології виявлення шахрайства необхідно розробити концептуальну модель інформаційної технології виявлення шахрайства. Розробимо таку концептуальну модель, яка в контексті технології UML [90], [91] описується за допомогою діаграм варіантів використання (use case diagram).

На рис. 3.1 наведена концептуальна модель інформаційної технології. Щодо інформаційної технології виявлення шахрайства визначено 4 актори:

- користувач інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків – адміністратор;
- користувач мобільного додатку (органічний або шахраський);
- вебсервер, що обслуговує запити до мобільного додатку;
- інформаційна технологія виявлення шахрайства при інсталюванні мобільних додатків.

Користувачі інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків реєструються у даній вебсистемі та можуть переглядати списки органічних і шахрайських користувачів та маркетингових кампаній.

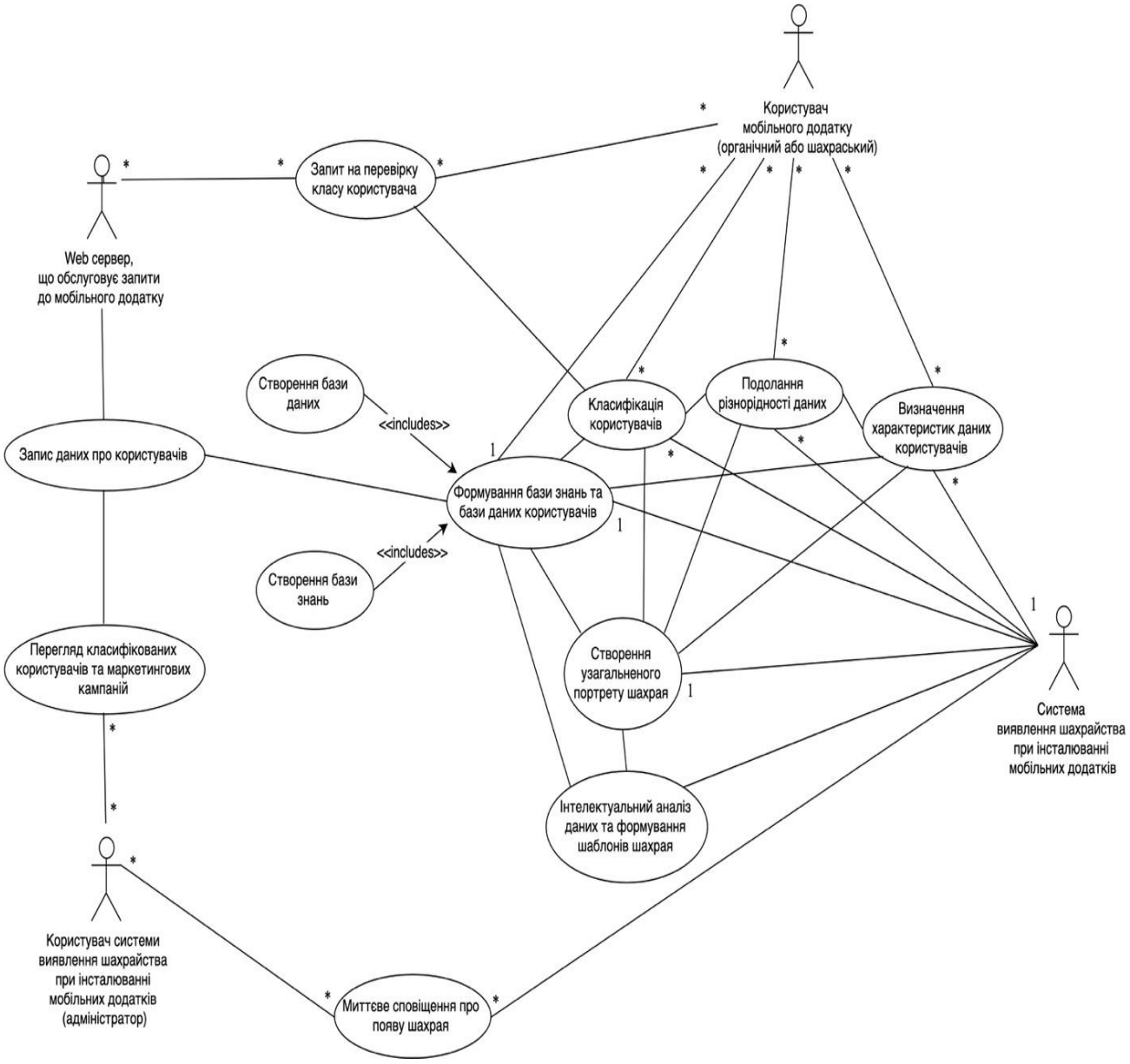


Рисунок 3.1 – Концептуальна модель інформаційної технології

Користувачі мобільного додатку, які можуть бути органічними або шахрайськими, яких і необхідно класифікувати, інсталюють мобільний додаток та здійснюють усі доступні їм дії у додатку.

Вебсервер, що обслуговує запити до мобільного додатку, записує дані про користувачів, а саме – всю інформації про їх дії, а також посилає запит до інформаційної технології виявлення шахрайства на здійснення класифікації даного користувача.

Розроблена інформаційна технологія виявлення шахрайства при інсталюванні мобільних додатків виконує такі функції:

– здійснює визначення характеристик даних користувачів;

– здійснює подолання різнорідності даних;

– здійснює класифікацію користувачів;

– формує базу знань та базу даних користувачів;

– створює узагальнений портрет шахрая;

– здійснює інтелектуальний аналіз даних та формування шаблонів шахрая;

– здійснює миттєве сповіщення про появу шахрая у випадку, якщо по своїй новій дії користувач буде визначений шахрайським.

При появі нової події користувача (нового або вже існуючого), інформаційна технологія виявлення шахрайства визначає клас даного користувача. Якщо користувач виявляється шахрайським, то інформаційна технологія сповіщає про це усіх адміністраторів.

На рис. 3.2 зображена логічна модель інформаційної технології виявлення шахрайства (class Diagram), яка лежить в основі реалізацій [92]-[94]. Блоки – це програмні сутності інформаційної технології (класи), які реалізують певну функціональність. Кожен блок складається з двох секцій – назви класу та переліку виконуваних операцій.

В процесі декомпозиції інформаційної технології були отримані такі структурні одиниці:

– підсистема виявлення характеристик користувача;

– підсистема подолання різнорідності даних;

– підсистема побудови класифікаційної моделі;

– підсистема класифікації;

– підсистема формування бази даних шахраїв;

– підсистема формування бази знань (для виявлення шахраїв);

– підсистема інтелектуального аналізу даних та формування портрету користувачів;

– підсистема прогнозування узагальненого шаблону шахрая.

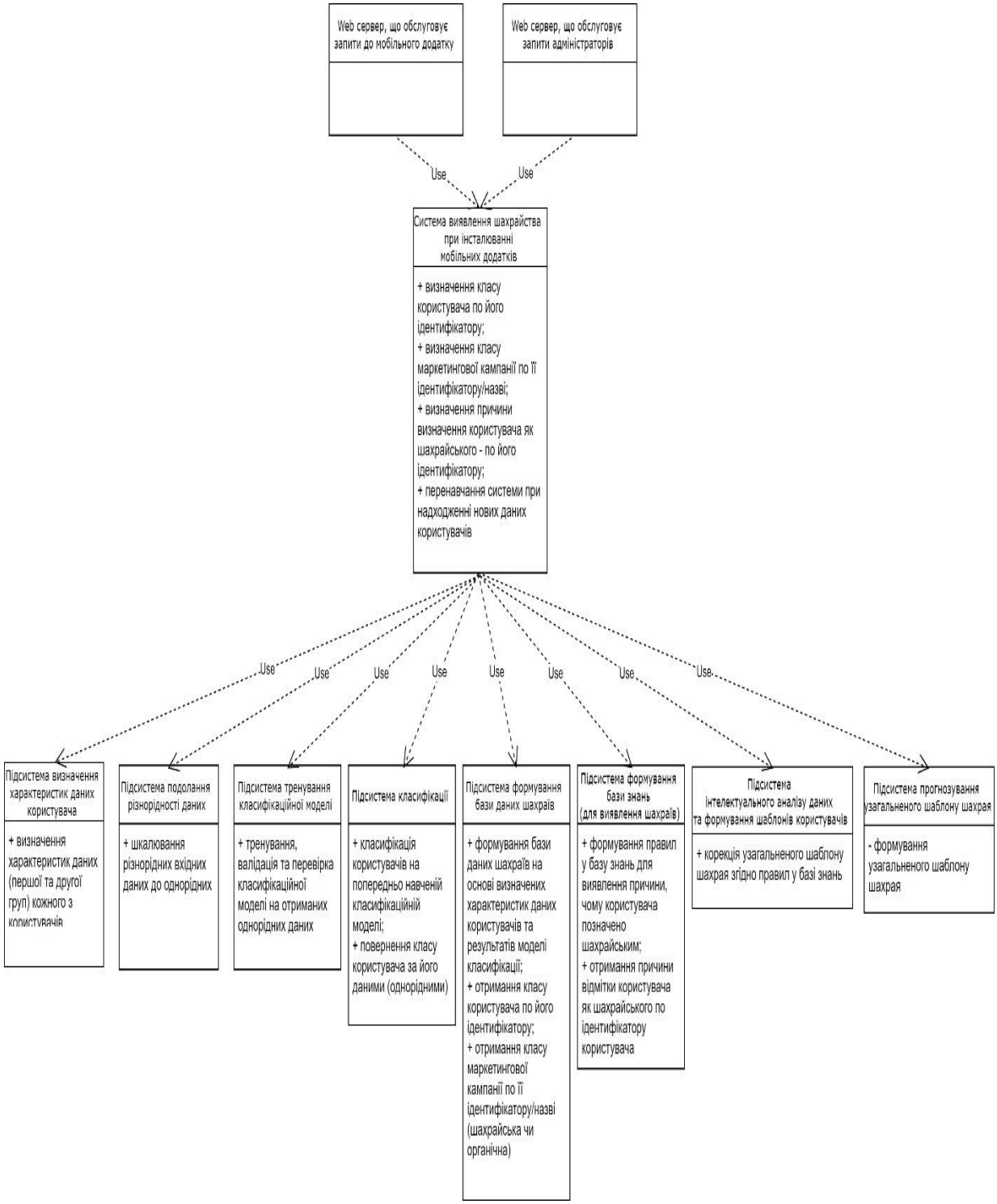


Рисунок 3.2 – Логічна модель інформаційної технології

Зазначені підсистеми формують функціональну основу інформаційної технології виявлення шахрайства. Порядок, у якому іде звернення до підсистем, подано на рис. 3.3 [18].

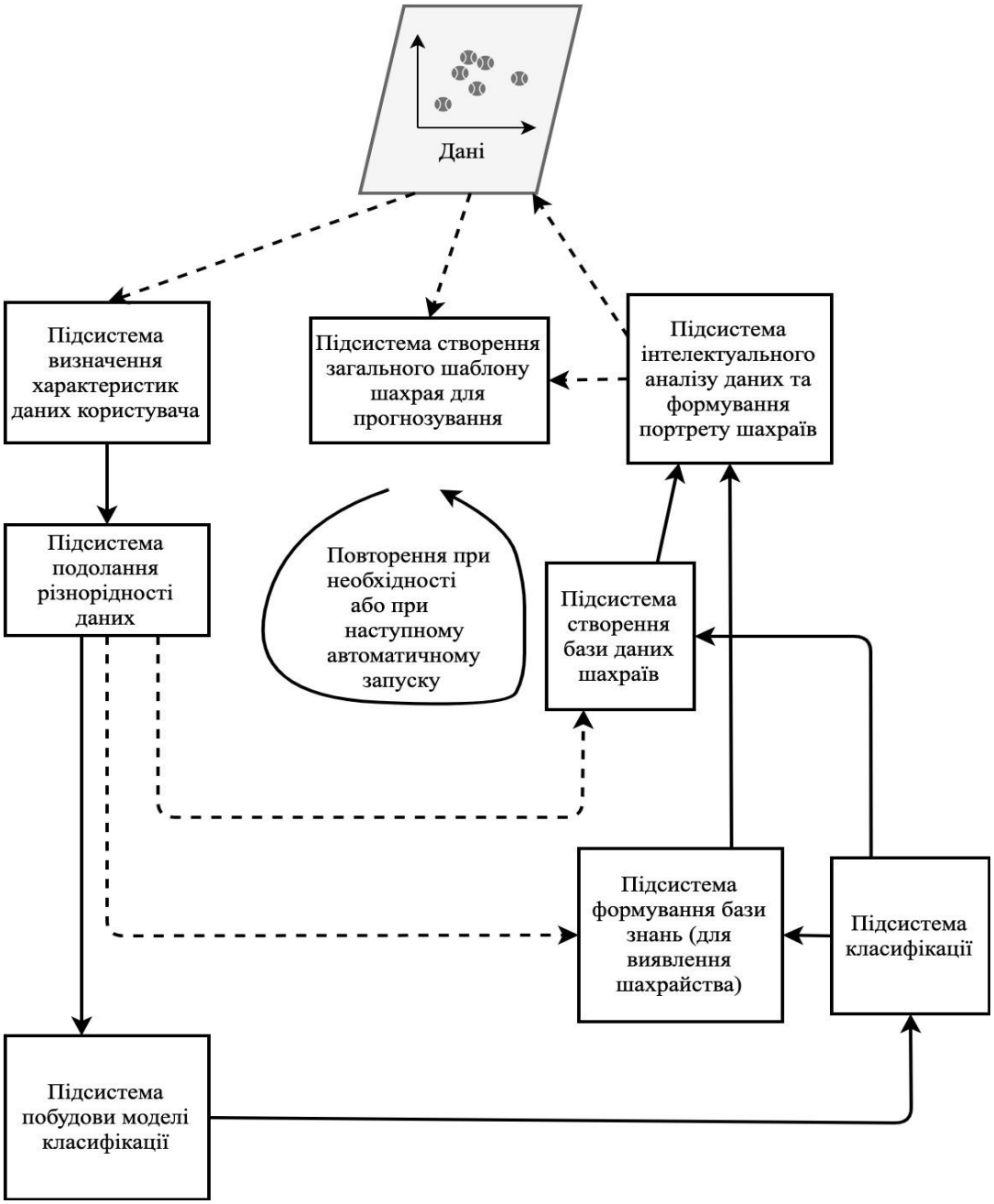


Рис. 3.3. Структура інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків

3.1.2 Аналіз та розробка структур даних в інформаційній технології виявлення шахрайства

Спроектуємо бази даних і перед початком проектування бази даних розглянемо природу даних, які необхідно зберігати:

– необхідно зберігати мільярди записів у базі даних;

– кожен об’єкт, який необхідно зберігати, невеликий, менший ніж 1 Кб;

– між записами немає зв’язків, окрім збереження користувача, який здійснив дану подію;

– інформаційна технологія підтримує високе навантаження на читання даних (read-heavy).

Базуючись на інформації про природу даних, запропонуємо схему бази даних (рис. 3.4).

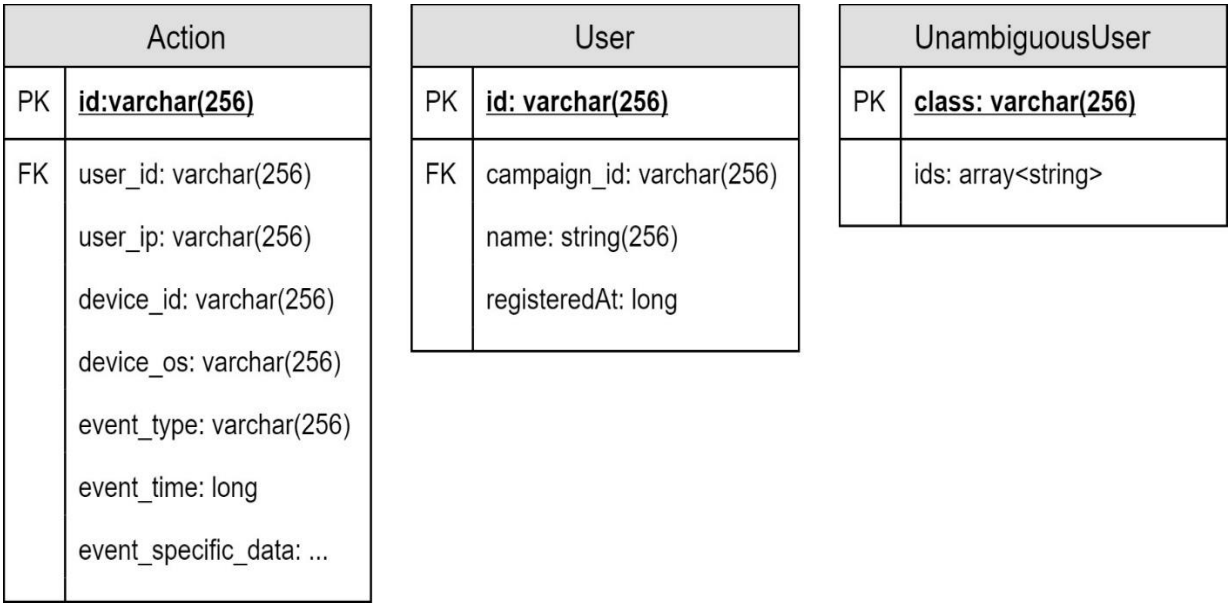


Рисунок 3.4 – Схема бази даних інформаційної технології виявлення шахрайства

Для вибору підходящої бази даних для представленої на рис. 3.4 схеми, розглянемо різницю між існуючими. Та різницю між реляційними і нереляційними базами даних для коректного вибору бази даних.

Реляційні бази даних зберігають дані у рядках та колонках таблиць. Кожен рядок містить всю інформацію про один запис (об’єкт), а кожна колонка містить усю інформацію по певному полю об’єкта. Найвідомішими реляційними базами даних є Postgres, MySQL, MS SQL Server, Oracle, SQLite, MariaDB.

Нереляційні бази даних бувають різних типів та мають особливості в залежності відповідного типу:

– документо-орієнтована система керування базами даних (document database). У базах даних такого типу дані зберігаються у документах, замість рядків та колонок в таблицях, що використовуються у реляційних базах даних. Такі документи групуються разом у колекції. Кожен з документів однієї і тієї ж колекції може мати повністю різну структуру, оскільки всі поля не є обов'язковими, як це є у реляційних базах даних. Відомими базами даних такого типу є MongoDB, CouchDB;

– бази даних типу «ключ-значення» (key-value stores). Дані зберігаються у масивах пар типу «ключ-значення». Ключем у базі даних такого типу є назва атрибуту, який пов'язаний із «значенням». Відомими базами даних такого типу можна відзначити Redis, Dynamo, Voldemort тощо;

– графова база даних (graph databases). Бази даних такого типу зберігають ті дані, взаємозв'язки між якими найкраще подаються за допомогою графу. Дані зберігаються у такій структурі даних як граф з вершинами (об'єктами – entities), властивостями (інформацією про об'єкти), та ребрами графу (зв'язки між об'єктами). Відомими базами даних такого типу є InfiniteGraph, Neo4J тощо;

– сховище з широким стовпчиком (wide-column databases). У базах даних такого типу замість таблиць, що використовуються в реляційних базах даних, використовуються «сімейства» колонок, які є контейнерами для рядків. Проте, на відміну від реляційних баз даних, тут кожен рядок не повинен мати однакову кількість колонок. Такий тип баз даних найкраще підходить для аналізу великих наборів даних. Відомими базами даних такого типу є Cassandra, HBase.

Розглянувши основні типи баз даних, розглянемо відмінності між ними на високому рівні проектування, а саме – розглянемо відмінності між SQL та NoSQL базами даних, як їх прийнято називати:

– схема. В SQL базах данних кожен запис відповідає фіксованій схемі. Тобто колонки необхідно сформувати перед введенням даних, і кожен рядок повинен мати дані для кожної колонки. Схема може бути змінена пізніше, але

вона включає зміну всієї бази даних і перехід в автономний режим при цьому. В NoSQL схеми динамічні. Колонки можна додавати динамічно, і кожен запис (еквівалент рядку у реляційних базах даних) не повинен містити дані для кожної колонки;

– запити. SQL бази даних використовують мову структурованих запитів SQL (structured query language) для визначення та маніпулювання даними, що є дуже об’ємними. У NoSQL базах даних запити працюють з колекціями документів. Їх часто називають UnQL (unstructured query language – мова неструктурованих запитів). Різні бази даних мають різний синтаксис для використання UnQL;

– масштабування. У більшості випадків SQL бази даних вертикально масштабовані, тобто відбувається за рахунок збільшення потужності апаратного забезпечення (більш висока пам'ять, процесор і т.д.), яке може стати дуже дорогим. Можна масштабувати реляційні бази даних на декількох серверах, але це складний і трудомісткий процес. З іншого боку, NoSQL бази даних є горизонтально масштабованими, тобто існує можливість легко додати більше серверів в інфраструктуру NoSQL бази даних для обробки великого трафіку. Будь-яке дешеве обладнання або хмарні екземпляри можуть розміщувати NoSQL бази даних, що робить горизонтально масштабування більш рентабельним, ніж вертикальне масштабування. Багато технологій NoSQL також автоматично поширюють дані між серверами;

– надійність або відповідність ACID (atomicity – атомарність, consistency – узгодженість, isolation – ізольованість, durability – довговічність). Переважна більшість реляційних баз даних є сумісними з ACID. Тому, коли йдеться про надійність даних і безпечну гарантію виконання транзакцій, SQL-бази даних є найкращою опцією. Більшість рішень NoSQL жертвують дотриманням ACID задля продуктивності і масштабованості.

Зазначимо, що при виборі бази даних не є однозначно правильного рішення. Тому багато компаній використовують як реляційні, так і нереляційні бази даних в залежності від різних потреб. Навіть зважаючи на те, що NoSQL

бази даних набирають популярність завдяки своїй швидкості та масштабованості, все ще існують ситуації, коли високо структурована SQL база даних працює краще. Вибір правильної технології залежить від сфери використання.

Так, реляційну (або ж SQL) базу даних обирають, коли:

– необхідно забезпечити дотримання ACID. Відповідність ACID знижує аномалії та захищає цілісність бази даних, прописуючи, як саме операції взаємодіють з базою даних. Як правило, NoSQL бази даних жертвують дотриманням ACID для забезпечення масштабованості та швидкості обробки, але для багатьох комерційних і фінансових програм база даних, що відповідає ACID, залишається найкращою опцією;

– якщо існуючі дані структуровані і незмінні, та якщо не відбувається значного зростання кількості даних, що потребує більшої кількості серверів, і якщо система працює лише з узгодженими даними, немає жодної причини використовувати систему, призначену для підтримки різноманітних типів даних і великого обсягу трафіку.

Нереляційну (або ж NoSQL) базу даних обирають, коли:

– необхідне зберігання великих обсягів даних, які часто не мають структури. NoSQL база даних не встановлює обмежень на типи даних, які можна зберігати разом, і дозволяє додавати нові типи, як необхідні зміни. У базах даних на основі документів можна зберігати дані в одному місці без необхідності заздалегідь визначати, які "типи" даних повинні бути.

– необхідне оптимальне використання хмарних обчислень і зберігання. Хмарне сховище є відмінним рішенням, яке дає змогу заощадити витрати, але потребує, щоб дані легко поширювались між декількома серверами. Використання товарного (доступного, меншого) обладнання на місці або в хмарі дозволяє уникнути встановлення додаткового платного програмного забезпечення, і NoSQL бази даних, такі як Cassandra, розраховані на масштабування в декількох центрах обробки даних.

– необхідна підтримка швидкого розвитку системи / бази даних / об’єктів. NoSQL є надзвичайно корисним при швидкому розвитку записів бази даних, оскільки їх не потрібно попередньо оновлювати. Якщо існують швидкі ітерації системи, які вимагають частого оновлення структури даних без великої кількості простоїв між версіями, реляційна база даних буде сповільнювати робочий процес.

Тобто, коли всі компоненти системи працюють швидко і бездоганно, NoSQL бази даних запобігають тому, щоб дані були вузьким місцем (bottleneck). Великі дані (big data) посприяли великому успіху NoSQL баз даних, головним чином тому, що вони обробляють дані не так як традиційні реляційні бази даних. Кілька популярних прикладів NoSQL баз даних – MongoDB, CouchDB, Cassandra і HBase.

То яку ж базу даних обрати для розробки інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків? На думку авторів, оскільки необхідно зберігати мільярди даних, а взаємозв’язків між об’єктами немає, то найкращим рішенням буде використовувати нереляційну NoSQL базу даних типу «ключ-значення», таку як DynamoDB, MongoDB, Cassandra або Riak. Також, нереляційну базу даних буде легше масштабувати. У даній десертаційній роботі перевагу було надано нереляційній базі даних MongoDB.

3.1.3 Розробка шаблону шахрая

В процесі обробки даних необхідною є обробка початкового узагальненого шаблону шахрая. Для його розробки було проведено ряд експериментів з використанням даних з мобільних додатків компаній, у яких було здійснено акт впровадження (Додаток А). По результатам досліджень на даних продуктах було отримано вектори, які характеризують шаблон шахрая. Після аналізу даних векторів експертами створено вектор контрольних параметрів, що заданий у таблиці 3.1. Зазначимо, що шаблон складають лише характеристики, які входять у другу групу G_2 згідно до пункту 2.3.1.

3.1.4 Розробка нечіткої моделі для формування портрету шахрая

Як було розглянуто у пунктах 1.2.1 та 2.3.3, метод класифікації дозволяє дати чітку відповідь: користувач шахрай чи органічний. Запропонована інформаційна технологія, у свою чергу, дасть змогу визначати причину, по якій користувач був помічений шахраєм. А також покаже, коли необхідно перенавчити модель класифікації. Така можливість буде завдяки розробці нечіткої моделі для формування портрету шахрая. Розглянемо її.

В процесі обробки даних, а саме, після вирішення задач визначення характеристик про користувача, подолання різнорідності даних та класифікації є всі наявні дані для формування портрету шахрая.

Отже, звернемо увагу на те, що нечіткість вхідних показників, які характеризують користувачів мобільного додатку і недосконаліть існуючих математичних моделей для використання інших методів дають право стверджувати, що використання нечіткої логіки також є доцільним при формуванні нових правил визначення груп шахрайських та органічних користувачів для запису їх в базу знань, на основі яких здійснюватиметься визначення коефіцієнтів групи $\overline{G2}$ та подальша класифікація. Тому з метою вирішення даної задачі використаємо систему з нечіткою логікою, подавши вхідну інформацію нечіткими множинами $\tilde{B}_1$ на універсальній множині U, які подаватимемо сукупностями пар

$$\left(\mu_{\tilde{B}_1}(u), u\right), \tag{3.1}$$

де $\mu_{\tilde{B}_1}(u)$ – ступінь належності елемента $u \in U$ до нечіткої множини $\tilde{B}_1$. Ступінь належності знаходиться в діапазоні [0, 1]. Чим вище ступінь належності, тим у більшій мірі елемент універсальної множини U відповідає властивостям нечіткої множини [4].

Таблиця 3.1

Вектор контрольних параметрів

Назва параметру	Контрольний вектор
Кількість кліків з одного пристрою за хвилину	$\{time_1, time_2, time_3, ...\}$ – залежить від додатку (у кожного свої анімації, розмір картинок тощо)
Час між подіями користувача	$\{time\_ac_1, time\_ac_2, time\_ac_3, ...\}$– залежить від додатку (у кожного свої анімації, розмір картинок тощо)
Час між подіями користувача (перевірка на бота) за певний проміжок часу	$\{time\_bot_1, time\_bot_2, time\_bot_3, ...\}$ – залежить від додатку (у кожного свої анімації, розмір картинок тощо)
Тип подій користувача – скільки серед доступних він використовує	$\{type_1, type_2, ...\}$– залежить від додатку (у кожного свої типи подій)
Частота кожного типу події користувача (час між подіями кожного типу)	$type_1 : \{time_1, time_2, time_3, ...\}$ – залежить від додатку (у кожного свої анімації, розмір картинок тощо) $type_2 : \{time_1, time_2, time_3, ...\}$– залежить від додатку (у кожного свої анімації, розмір картинок тощо) Множина типів подій залежить від додатку (у кожного свої типи подій)
Соціальні мережі – кількість друзів у соціальній мережі (якщо підв'язаний до соціальної мережі)	$[N_{min}; N_{max}]$
Кількість акаунтів з одного пристрою	$[Ac_{min}; Ac_{max}]$
Кількість інсталювань з одного пристрою	$[K_{min}; K_{max}]$
Час між інсталюваннями на одному пристрої	$\{time\_d_1, time\_d_2, time\_d_3, ...\}$
Час інсталювання	$[I\_Time_{min}; I\_Time_{max}]$
Кількість кліків з однієї IP-адреси за хвилину	$[Kip_{min}; Kip_{max}]$

Вплив факторів

$$X = \{x_1, x_2, ..., x_n\} \tag{3.2}$$

на значення кожного з параметрів визначимо з використанням нечіткої бази знань, яку задамо у вигляді логічних висловлювань наступного типу:

$$\begin{aligned}
&\text{Якщо} \quad i_1 = b_1^{j1} \qquad i_2 = b_2^{j1} \qquad \ldots \text{ та } i_n = b_n^{j1}, \quad \text{або} \\
&\qquad\quad i_1 = b_1^{j2} \qquad i_2 = b_2^{j2} \qquad \ldots \text{ та } i_n = b_n^{j2}, \quad \text{або} \\
&\qquad\quad i_1 = b_1^{jk_j} \qquad i_2 = b_2^{jk_j} \qquad \ldots \text{ та } i_n = b_n^{jk_j}, \\
&\qquad\qquad \text{то} \quad y = c_j, j = \overline{1,m}, \qquad (3.3)
\end{aligned}$$

де b_1^{jp} – лінгвістичний терм, що оцінює значення фактора x_1 в p-*ій* диз'юнкції *j-го* логічного висловлювання

$$\left(j = \overline{1,m}, p = \overline{1,k}, x = \overline{1,n}\right); \qquad (3.4)$$

k_j – число диз'юнкції (або) в j-*му* логічному висловлюванні.

Прийняття рішення здійснюватиметься на основі визначення значень вхідних параметрів, а це означає, що алгоритм прийняття рішення має такий загальний вигляд «ЯКЩО умова1 … умова2 … ТО». З цього можна зробити висновок, що найбільш сумісною моделлю представлення знань для обраної предметної області є продукційні правила. У даному випадку продукційні правила будуть побудовані на нечітких множинах, що полегшить прийняття рішень в області виявлення аномалій при інсталюванні мобільних додатків.

Апроксимувавши залежності

$$y = f\left(x_1, x_2, ..., x_n\right) \qquad (3.5)$$

за допомогою нечіткої бази знань та операцій над нечіткими множинами, отримуємо нечіткий логічний висновок [4], [105], [106].

Для початку визначимо вхідні параметри системи, чітка ідентифікація яких подається на вхід системи (табл. 3.2). Усі вхідні характеристики було визначено у підрозділі 2.3, їх детальний опис представлено у таблиці 2.1.

Таблиця 3.2

Набір вхідних лінгвістичних параметрів системи *X*

Назва вхідного параметра	Позначення
Кількість кліків з одного пристрою за хвилину	x_1
Час між подіями користувача	x_2
Час між подіями користувача (перевірка на бота) за певний проміжок часу	x_3
Тип подій користувача – скільки серед доступних він використовує	x_4
Частота кожного типу події користувача (час між подіями кожного типу)	x_5
Соціальні мережі – користувач підв’язаний до соціальної мережі чи ні	x_6
Соціальні мережі – кількість друзів у соціальній мережі (якщо підв'язаний до соціальної мережі)	x_7
Кількість акаунтів з одного пристрою	x_8
Кількість інсталювань з одного пристрою	x_9
Час між інсталюваннями на одному пристрої	x_{10}
Час інсталювання	x_{11}
Інформація про користувача	x_{12}
Фото користувача	x_{13}
Кількість кліків з однієї IP-адреси за хвилину	x_{14}
Ідентифікатор пристрою користувача	x_{15}
Покупки – куплена / не куплена	x_{16}

Продовження таблиці 3.2

Назва вхідного параметра	Позначення
Покупки – куплена доступна чи недоступна	x_{17}
Покупки – підтверджена / не підтверджена	x_{18}
Верифікація IP-адреси: входить до списку шахрайських чи ні	x_{19}
Тип події – доступна / недоступна	x_{20}

Також зазначимо параметр під назвою «Клас», який користувач отримає на виході системи (табл. 3.3).

Таблиця 3.3

Набір вихідних параметрів системи *Y = Q*

Назва вихідного параметра	
Органічний користувач	q_1
Шахрай	q_2

Необхідно реалізувати механізм, що формує з чіткого параметру вхідного значення нечітке. Цей процес називається фазифікацією.

Сформуємо набір вхідних та вихідних параметрів системи:

$$X' = \left[x_1', x_2', x_3', x_4', x_5', x_6', x_7', x_8', x_9', x_{10}', x_{11}', x_{12}', x_{13}', x_{14}', x_{15}', x_{16}', x_{17}', x_{18}', x_{19}', x_{20}'\right] \quad (3.6)$$

$$Q' = \left[q_1', q_2'\right] \quad (3.7)$$

Подальшим кроком визначаємо терми для кожної лінгвістичної змінної (табл. 3.4).

Для параметрів «кількість кліків з одного пристрою за хвилину», «частота кожного типу події користувача (час між подіями кожного типу)», «кількість кліків з однієї IP-адреси за хвилину» доцільно ввести три лінгвістичних терми –

«низька», «середня», «висока», оскільки ці параметри, на відміну від такого параметру як, наприклад, «соціальні мережі – кількість друзів у соціальній мережі (якщо підв'язаний до соціальної мережі)», завжди присутні у кожного користувача. Для вихідного параметру «клас користувача» доцільно ввести два терми – «шахрай», «органічний користувач». Дані по всім параметрам представлено у таблиці 3.4.

Наступним кроком є опис і обґрунтування *ч*-функції належності для кожного терма. Найбільш розповсюджені методи побудови функцій належностей ґрунтуються на статистичній обробці експертної інформації та на парних порівняннях [56]-[58], [107].

Зручним представленням функцій належності є представлення у параметричній формі – має 2, 3 або 4 параметри. Існує багато типів параметричних функцій належностей, найбільш розповсюдженими серед яких є трикутна, трапецієвидна та дзвіноподібна.

Таблиця 3.4

Опис лінгвістичних змінних нечіткої системи

Лінгвістична змінна	**Терм-множина**	**Символьне представлення (відповідно)**
Кількість кліків з одного пристрою за хвилину	T_1=*{низька, середня, висока}*	$T_1 = \{A,B,C\}$
Час між подіями користувача	T_2=*{малий, середній, високий}*	$T_2 = \{A,B,C\}$
Час між подіями користувача (перевірка на бота) за певний проміжок часу	T_3=*{малий, середній, високий}*	$T_3 = \{A,B,C\}$
Тип подій користувача – скільки серед доступних він використовує	T_4=*{малий, середній, високий}*	$T_4 = \{A,B,C\}$
Частота кожного типу події користувача (час між подіями кожного типу)	T_5=*{низька, середня, висока}*	$T_5 = \{A,B,C\}$
Соціальні мережі – чи підв’язаний користувач до соціальної мережі	T_6=*{ ні, так}*	$T_6 = \{A,B\}$

Продовження таблиці 3.4

Лінгвістична змінна	Терм-множина	Символьне представлення (відповідно)
Соціальні мережі – кількість друзів у соціальній мережі (якщо підв'язаний до соціальної мережі)	T_7={ *мала, середня, велика, близька до максимуму}*	$T_7 = \{A,B,C,D\}$
Кількість акаунтів з одного пристрою	T_8={ *один, мала, середня, висока}*	$T_8 = \{A,B,C,D\}$
Кількість інсталювань з одного пристрою	T_9={ *одне, мала, середня, висока}*	$T_9 = \{A,B,C,D\}$
Час між інсталюваннями на одному пристрої	T_{10}={*малий, середній, високий}*	$T_{10} = \{A,B,C\}$
Час інсталювання	T_{11}={*малий, середній, високий}*	$T_{11} = \{A,B,C\}$
Інформація про користувача	T_{12}={ *майже співпадає з інформацією про існуючого користувача, мінімальне співпадіння з інформацією про існуючих користувачів}*	$T_{12} = \{A,B\}$
Фото користувача	T_{13}={ *майже співпадає з фото існуючого користувача, мінімальне співпадіння з фото існуючих користувачів}*	$T_{13} = \{A,B\}$
Кількість кліків з однієї IP-адреси за хвилину	T_{14}={*низька, середня, висока}*	$T_{14} = \{A,B,C\}$

Продовження таблиці 3.4

Лінгвістична змінна	Терм-множина	Символьне представлення (відповідно)
Ідентифікатор пристрою користувача	T_{15}={ *входить до списку шахрайських, не входить до списку шахрайських* }	$T_{15} = \{A, B\}$
Покупки – куплена / не куплена	T_{16}={ купував , не купував }	$T_{16} = \{A, B\}$
Покупки – куплена доступна чи недоступна	T_{17}={ *купував недоступні покупки, купував доступні покупки*}	$T_{17} = \{A, B\}$
Покупки – підтверджена / не підтверджена	T_{18}={ *відправляв запити про непідтверджені покупки, відправляв запити лише про підтверджені покупки* }	$T_{18} = \{A, B\}$
Верифікація IP-адреси: входить до списку шахрайських чи ні	T_{19}={ *входить до списку шахрайських IP-адрес, не входить до списку шахрайських IP-адрес* }	$T_{19} = \{A, B\}$
Тип події – доступна / недоступна	T_{20}={ *відправляв запити на недоступні типи подій, відправляв запити лише на доступні типи подій* }	$T_{20} = \{A, B\}$
КЛАС КОРИСТУВАЧА	$T_{вих.}$={ *шахрай, органічний користувач*}	$T_{вих.} = \{A, B\}$

Трикутна модель функції належності потребує трьох параметрів, якими є координати максимуму та мінімумів. Трапецієвидна модель функції належності потребує чотирьох параметрів, якими є координати максимумів та мінімумів. Дзвіноподібна модель функції належності потребує двох параметрів, якими є координата максимуму (b) та коефіцієнт концентрації функції належності (c).

Дзвіноподібна модель функції належності має лише 2 параметри, тому при її використанні зменшується розмірність оптимізаційної задачі, яка виникає при реалізації нечіткої моделі. Дзвіноподібна модель функції забезпечує достатню "гнучкість" представлення нечіткої інформації.

При побудові нечіткої системи вибір функції належності та значення коефіцієнтів має вирішальну роль. Саме від функції належності та її параметрів залежить адекватність системи і її працездатність.

Для формалізації лінгвістичних змінних оберемо дзвіноподібну модель функції належності, яка має найменше число параметрів, що зменшує розмірність задачі підбору цих параметрів при реалізації нечіткої моделі.

Аналітичну модель функції належності змінної x до довільного нечіткого терму T представимо у вигляді (3.8).

$$\mu^T(x) = \frac{1}{1 + \left(\frac{x-b}{c} \right)^2}, \tag{3.8}$$

де b – координата максимуму функції, $\mu^T(b) = 1$;

c – коефіцієнт концентрації-розтягування функції.

Фактично число b – це найбільш прагматичне значення змінної x для нечіткого терму T.

Для прикладу, графік належності параметру «кількість інсталювань з одного пристрою» зображено на рис. 3.5.

Задача експерта полягає у налаштування функції належності шляхом підбору коефіцієнтів, грунтуючись на деяких експериментальних даних. У

програмному засобі будуть встановлені коефіцієнти, надані експертом, і надана можливість змінювати їх вручну.

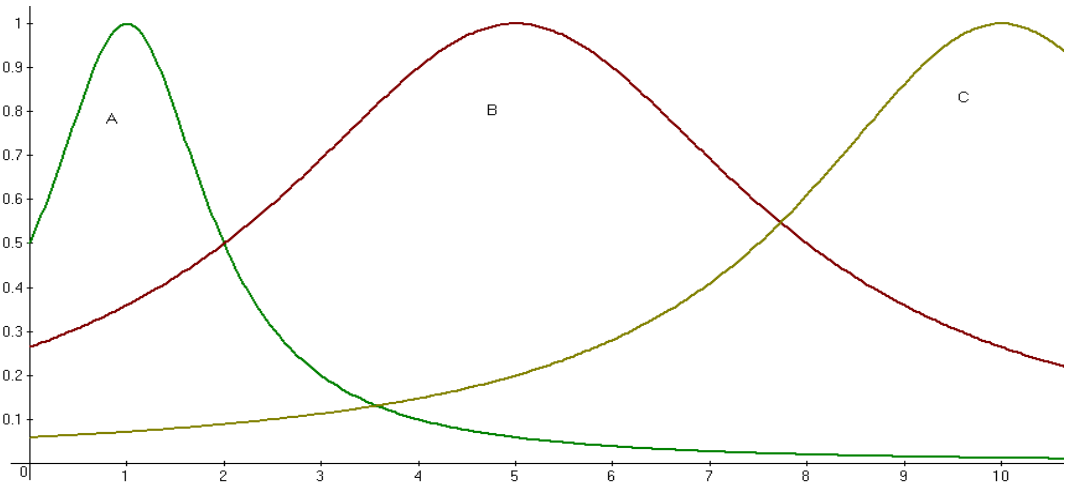


Рисунок 3.5 – Графік функції належності для параметру «кількість інсталювань з одного пристрою»

Відзначимо також допустимі границі вхідних значень для кожного входу. Всі вони будуть задаватись дійсними цілими числами, але в різних діапазонах. Діапазони обиратимуться на основі вхідних та вихідних даних моделі класифікації, що розглянута у підрозділі 2.3.3.

Зазначимо, що набір правил будувався на основі вхідних та вихідних даних моделі класифікації, що представлена у підрозділі 2.3.3. Тобто, маючи значення усіх вхідних параметрів системи та значення вихідного параметру, а саме – класу користувача, був побудований набір правил нечіткої логіки, що являє собою портрет шахрая. Слід зазначити, що проаналізувавши отримані правила, було виявлено спосіб їх оптимізації. Так, наприклад, якщо кількість акаунтів з одного пристрою користувача T_8 «висока» і при цьому, інформація про користувача T_{12} «майже співпадає з інформацією про існуючого користувача», то незалежно від значень інших параметрів даний користувач є шахраєм. Причому, можна однозначно сказати, що даний користувач є шахраєм по визначеним характеристикам x_8 та x_{12}. Слід зазначити, для того, щоб визначити клас користувача та причину помітки користувача як шахрая, достатньо знайти

відповідність характеристик користувача будь-якому з правил сформованого портрету шахрая. Але при необхідності повного аналізу дій користувача, є можливість здійснити перевірку належності характеристик користувача кожному із сформованих правил.

Матриця знань предметної області після її побудови, аналізу та оптимізації на основі здійсненого аналізу, представлена у таблиці 3.5. Зазначимо, що у таблиці 3.5 представлено матрицю знань предметної області для визначення Q_2 – шахрая, оскільки у даній роботі цікавить саме виявлення шахраїв. Кожен стовпець – це правило, що задає користувача, який відноситься до класу Q_2 .

Таблиця 3.5

Матриця знань предметної області

Q_j	Q_2																								
x_1													C		C	C	C	A		C	C		C		
x_2														C	C			A			C		C		
x_3																C		A	C	C			C		
x_4											B	C							A	A	A		A	A	A
x_5																						C		C	
x_6																									
x_7										A			A	A											
x_8														D											
x_9														D				D							D
x_{10}										C				C											
x_{11}											A	A													
x_{12}								A	A																
x_{13}							A		A																
x_{14}							C	C																	
x_{15}						A																			
x_{16}					A																				
x_{17}				A	A																				
x_{18}			A																						
x_{19}		A																							
x_{20}	A																								

Опишемо матрицю знань наступними рівняннями, які відображають правила «якщо - то»:

$$\mu^{Q_2}(x) = \mu^A(x_{20}) \vee$$
$$\mu^C(x_{14}) \cdot \mu^A(x_{13}) \vee$$
$$\mu^C(x_{10}) \cdot \mu^D(x_9) \cdot \mu^D(x_8) \cdot \mu^A(x_7) \cdot \mu^C(x_2) \vee$$
$$\mu^C(x_{14}) \cdot \mu^A(x_{12}) \vee$$
$$\mu^A(x_{13}) \cdot \mu^A(x_{12}) \vee$$
$$\mu^C(x_{10}) \cdot \mu^A(x_7) \vee$$
$$\mu^D(x_9) \cdot \mu^A(x_3) \cdot \mu^A(x_2) \cdot \mu^A(x_1) \vee$$
$$\mu^A(x_4) \cdot \mu^C(x_3) \vee$$
$$\mu^A(x_{11}) \cdot \mu^B(x_4) \vee$$
$$\mu^A(x_4) \cdot \mu^C(x_3) \cdot \mu^C(x_1) \vee$$
$$\mu^C(x_1) \cdot \mu^C(x_2) \cdot \mu^A(x_4) \vee$$
$$\mu^C(x_5) \vee$$
$$\mu^A(x_4) \cdot \mu^C(x_3) \cdot \cdot \mu^C(x_2) \cdot \mu^C(x_1) \vee$$
$$\mu^C(x_5) \cdot \mu^A(x_4) \vee$$
$$\mu^A(x_4) \vee$$
$$\mu^A(x_{11}) \cdot \mu^C(x_4) \vee$$
$$\mu^C(x_1) \cdot \mu^C(x_2) \vee$$
$$\mu^C(x_1) \cdot \mu^C(x_3) \vee$$
$$\mu^A(x_{19}) \vee$$
$$\mu^A(x_{18}) \vee$$
$$\mu^A(x_{17}) \vee$$
$$\mu^A(x_7) \cdot \mu^C(x_1) \vee$$
$$\mu^C(x_1) \vee$$
$$\mu^A(x_{17}) \cdot \mu^A(x_{16}) \vee \ \mu^A(x_{15}) \qquad (3.9)$$

Так наприклад, модель, що описує матрицю знань (3.9, табл. 3.5), дозволяє визначити чи користувач шахрай, використовуючи набір описаних правил.

Запропонована система нечітких правил використовується для побудови бази знань, за допомогою якої можна сформувати портрет шахрая, який включає і шаблон шахрая. Схема формування портрету шахрая зображена на рис. 3.6.

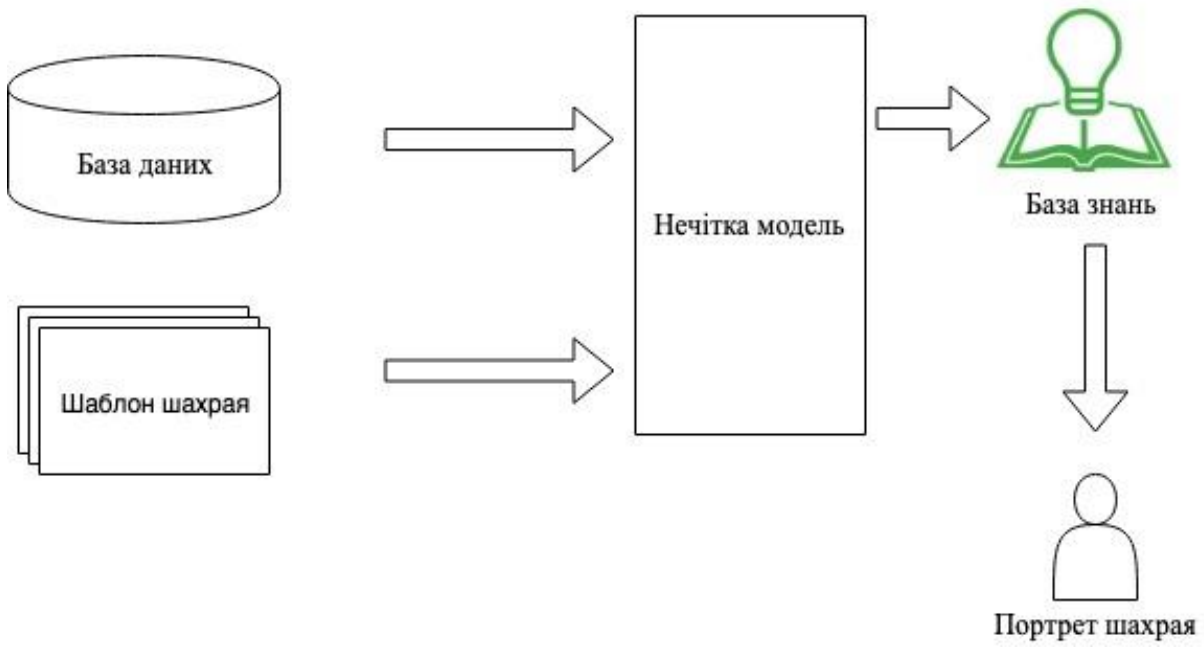


Рисунок 3.6 – Схема формування портрету шахрая

3.2 Розробка алгоритмів виявлення шахрая при інсталюванні мобільних додатків

Розробимо методику виявлення шахрайства та визначимо її. Методика включає в себе послідовне виконання таких основних алгоритмів::

– алгоритм 1 ініціалізації / корекції моделі виявлення шахрайства і бази знань;

– алгоритм 2 виявлення шахрайства при наявності ініціалізованої моделі виявлення шахрайства і бази знань.

Розглянемо дані алгоритми більш детально.

Алгоритм 1 ініціалізації / корекції моделі виявлення шахрайства та бази знань, що складається з алгоритмів 1.1 та 1.2.

Алгоритм 1.1:

1. Визначення характеристик даних користувачів по групам G_1 та G_2.

2. Створення узагальненого шаблону шахрая на основі поміченої вибірки, наданої експертами з використанням бази знань, що містить набір правил нечіткої логіки.

Алгоритм 1.2:

1. Подолання різнорідності вхідних даних користувачів:

1.1. Визначення коефіцієнтів за характеристиками групи G_1, з використанням яких також можна здійснити:

а) визначення однозначних органічних користувачів;

б) визначення однозначних шахрайських користувачів;

в) визначення неоднозначних користувачів.

1.2. Визначення коефіцієнтів за характеристиками групи G_2 з використанням коефіцієнтів подібності користувача за кожною з характеристик групи G_2 з узагальненим шаблоном шахрая.

1.3. Отримання вектору однорідних вхідних даних, а саме – сукупності конфіцієнтів, що представляють характеристики груп G_1 та G_2.

2. Тренування, валідація, тестування класифікаційної моделі, на вхід якої подаються однорідні дані. Тобто, побудова моделі класифікації, вхідними даними якої є однорідні дані, отримані з використанням узагальненого методу подолання різнорідності, розробленого в п. 2.3.

3. Коррекція узагальненого портрету шахрая на основі результатів з використанням моделі класифікації та вхідних однорідних даних моделі, отриманих з використанням методу подолання різнорідності даних (3.9, п. 3.1.4, табл. 3.5).

Алгоритм 2 виявлення шахрайства, при наявності ініціалізованої моделі виявлення шахрайства та бази знань, використовується для визначення класу нових користувачів або ж існуючих користувачів за їх новими діями у

мобільному додатку. Розглянемо кроки даного алгоритму при появі нової події нового або ж старого користувача у мобільному додатку:

1. Визначення характеристик користувача.

2. Визначення класу користувача з використанням бази знань, а саме – узагальненого шаблону користувача, за визначеними характеристиками.

3. У випадку, якщо існуюча база знань не дозволяє визначити клас поточного користувача, то виконати алгоритм 1.2 та повторити пункт 2 алгоритму 2.

Зазначимо, що наявність та використання узагальненого портрету шахрая дозволить автоматично виявляти шахраїв у будь-яких наборах даних, а також дасть змогу виявляти причину їх появи і здійснювати коригування узагальненого шаблону шахрая. Зважаючи на це, можна зазначити, що задача такого виду відноситься до задач нечіткою логіки.

Основною задачею модуля, який буде працювати з нечіткою логікою, є виявлення діапазону значень коефіцієнту подібності між набором характеристик користувача та узагальненим шаблоном шахрая по конкретній характеристиці, при потраплянні в який можна вважати користувача шахраєм по даній характеристиці (3.9, п. 3.1.4, табл. 3.5).

Формалізувавши проблему формування нових правил визначення груп шахрайських та органічних користувачів з використанням нечіткої логіки, доцільно запропонувати алгоритм автоматичного виявлення шахраїв:

1 Отримання списку користувачів довжиною *userCount* з бази даних.

2 Створення початкового узагальненого шаблону шахрая.

3 Встановлення значення кількості переглянутих алгоритмом користувачів *reviewedUsers* = 0.

4 *reviewedUsers* < *userCount*?

4.1 Поки умова 4 виконується, то виконуємо наступні дії:

4.1.1 Отримання вхідних даних по поточному користувачу.

4.1.2 Визначення коефіцієнтів групи 1 $\overline{G_1}$ по даному користувачу (пп. 2.3.1.2).

4.1.3 Хоча б один з коефіцієнтів визначає користувача шахраєм?

4.1.3.1 Якщо так, то:

a Помічення користувача шахраєм (віднесення до класу шахраїв Class0 та U_{fraud}).

b Формування нових правил відповідно формулам (2.13–2.25, 3.4 – 3.8) та додавання їх у базу знань.

4.1.3.2 Якщо ні, то: якщо хоча б один з коефіцієнтів визначає користувача органічним?

a) Помічення користувача органічним (Class1 and Uorg).

b) Формування нових правил відповідно формулам (2.13–2.25, 3.4 – 3.8) та додавання їх у базу знань.

4.1.4 Збільшуємо значення змінної *reviewedUsers* на 1.

4.2 Якщо умова 4 не виконується, то виконуємо наступні кроки:

4.2.1 Визначення вектору коефіцієнтів подібності користувача з шахрайськими користувачами $\overline{G}2_{fraud}$.

4.2.2 Визначення вектору коефіцієнтів подібності користувача з органічними користувачами $\overline{G}2_{org}$.

4.2.3 Формування спільного вектору характеристик

$$\overline{G}2 = \overline{G}2_{fraud} \cup \left(1 - \overline{G}2_{org}\right). \tag{3.10}$$

4.2.4 Формування нових правил відповідно формулам (2.13–2.25, 3.5 – 3.9) та додавання їх у базу знань.

4.2.5 Налаштування моделі класифікації з використанням даних помічених користувачів з вектору $\overline{G}2$ відповідно формулі

$$CV\left(\hat{f}\right) = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, \hat{f}^{-k(i)}\left(x_i\right)\right), \tag{3.11}$$

де N – кількість частин інформації;

L – функція втрат;

y – очікуваний результат для входу x;

$\hat{f}^{-k}(x)$ – фітнес-функція;

k – поточна частина інформації.

4.2.6 Визначення класів усіх користувачів на основі побудованої моделі класифікації.

4.2.7 Корегування узагальненого шаблону шахрая.

Запропонований алгоритм подано на рис. 3.7.

Алгоритм автоматичного виявлення шахраїв ліг в основу методу виявлення аномалій в різнорідних даних при інсталюванні мобільних додатків.

3.3 Розробка алгоритму створення узагальненого портрету шахрая

Розробимо алгоритм створення узагальненого портрету шахрая.

Слід зазначити, що узагальнений портрет шахрая задаватиметься системою правил нечіткої логіки (3.9, п. 3.1.4, табл. 3.5), що зберігатиметься у базі знань.

Як було зазначено у пп. 2.3.1.1, після визначення характеристик користувачів та коефіцієнтів по кожній з них, усі коефіцієнти можна розділити на дві групи. Перша група охоплює коефіцієнти, які дають змогу провести попередній аналіз даних, а саме, однозначно з множини користувачів визначити шахрайських користувачів, органічних та підозрілих. Коефіцієнти, що відносяться до першої групи представлено у пп. 2.3.1.1.

Друга група охоплює коефіцієнти, за якими неможливо зробити первинний аналіз. Коефіцієнти, що відносяться до другої групи представлено у пп. 2.3.1.1.

Узагальнений портрет шахрая містить набір правил нечіткої логіки для кожного з визначених експертами коефіцієнтів (3.9, табл. 3.5).

Для побудови узагальненого портрету шахрая у межах задачі виявлення шахрайства в роботі була використана система нечіткої логіки (п. 3.1.4). В основі розробленої системи – дані отримані від експертів.

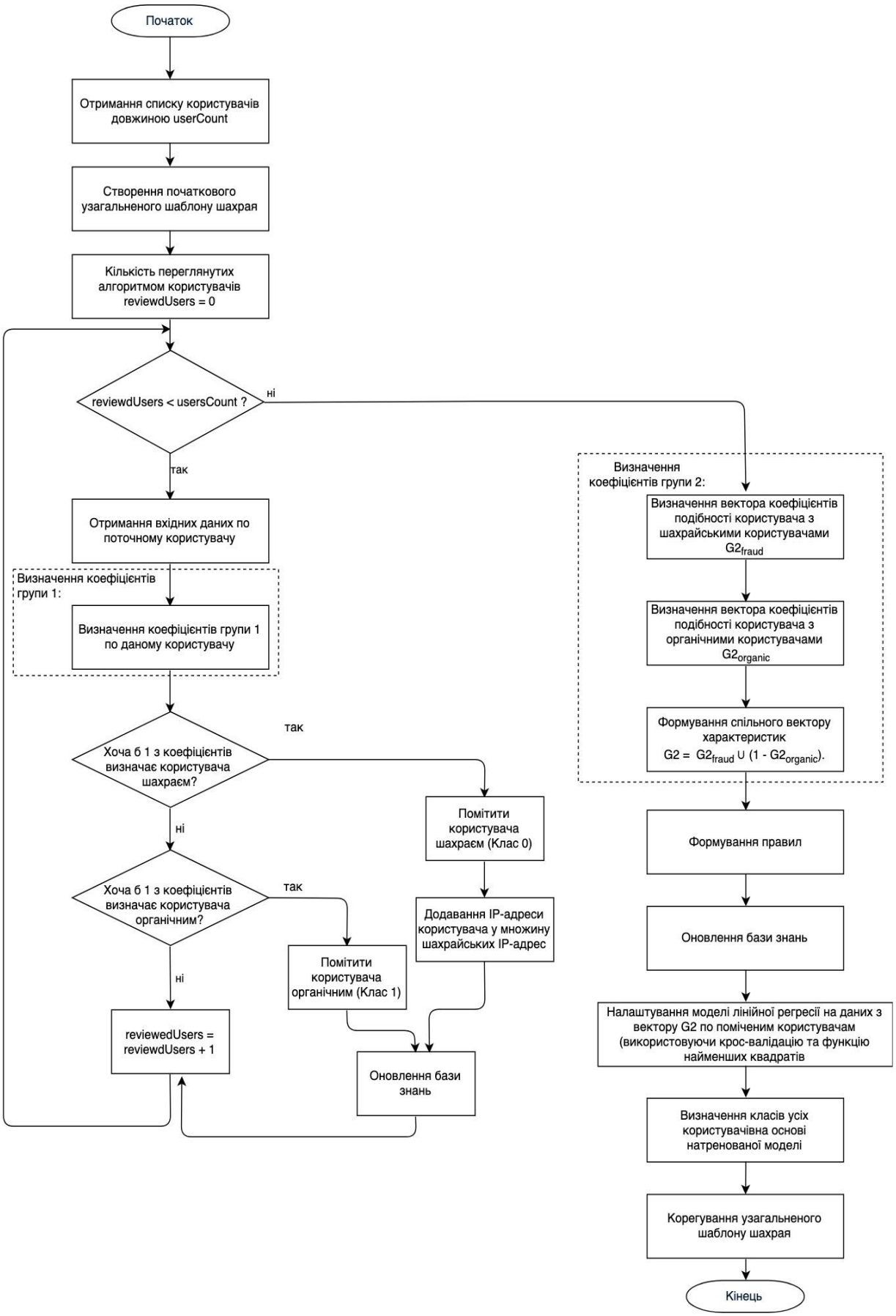


Рисунок 3.7 – Алгоритм автоматичного виявлення шахраїв

3.4 Розробка алгоритму пошуку аномалій в даних

Як було зазначено у пункті 3.1.1, вказані підсистеми формують функціональну основу інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків.

На рис. 3.8 зображена загальна модель функціонування інформаційної технології (Activity Diagram), робота якої ініціюється користувачем інформаційної технології після того, як він виконав певну дію – під час інсталювання або реінсталювання (повторного інсталювання), всередині або під час деінсталювання додатку. В результаті цього відбувається виконання послідовності дій процесу виявлення шахрайства при інсталюванні мобільних додатків відповідно до алгоритму функціонування інформаційної технології.

Розроблені концептуальні моделі інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків в результаті проведеного системного аналізу дають змогу здійснити як структурну, так і процесну декомпозиції інформаційної технології, що є необхідним кроком згідно зі стандартом ISO 12207. Наступним етапом є конкретизація реалізації інформаційної технології в термінах середовища функціонування та особливостей програмної імплементації.

3.5 Аналіз процесів в інформаційній технології виявлення шахрайства

Розглянемо детально кроки функціонування інформаційної технології виявлення шахрайства. Продукт допомагає виявити шахрайські маркетингові кампанії, які беруть оплату за інсталяції мобільних додатків. Отже, інформаційна технологія буде корисна бізнес-аналітикам.

Здійснимо аналіз процесів в інформаційній технології виявлення шахрайства при інсталюванні мобільних додатків за допомогою діаграми послідовності з використанням UML. На рис. 3.9 представлено діаграму

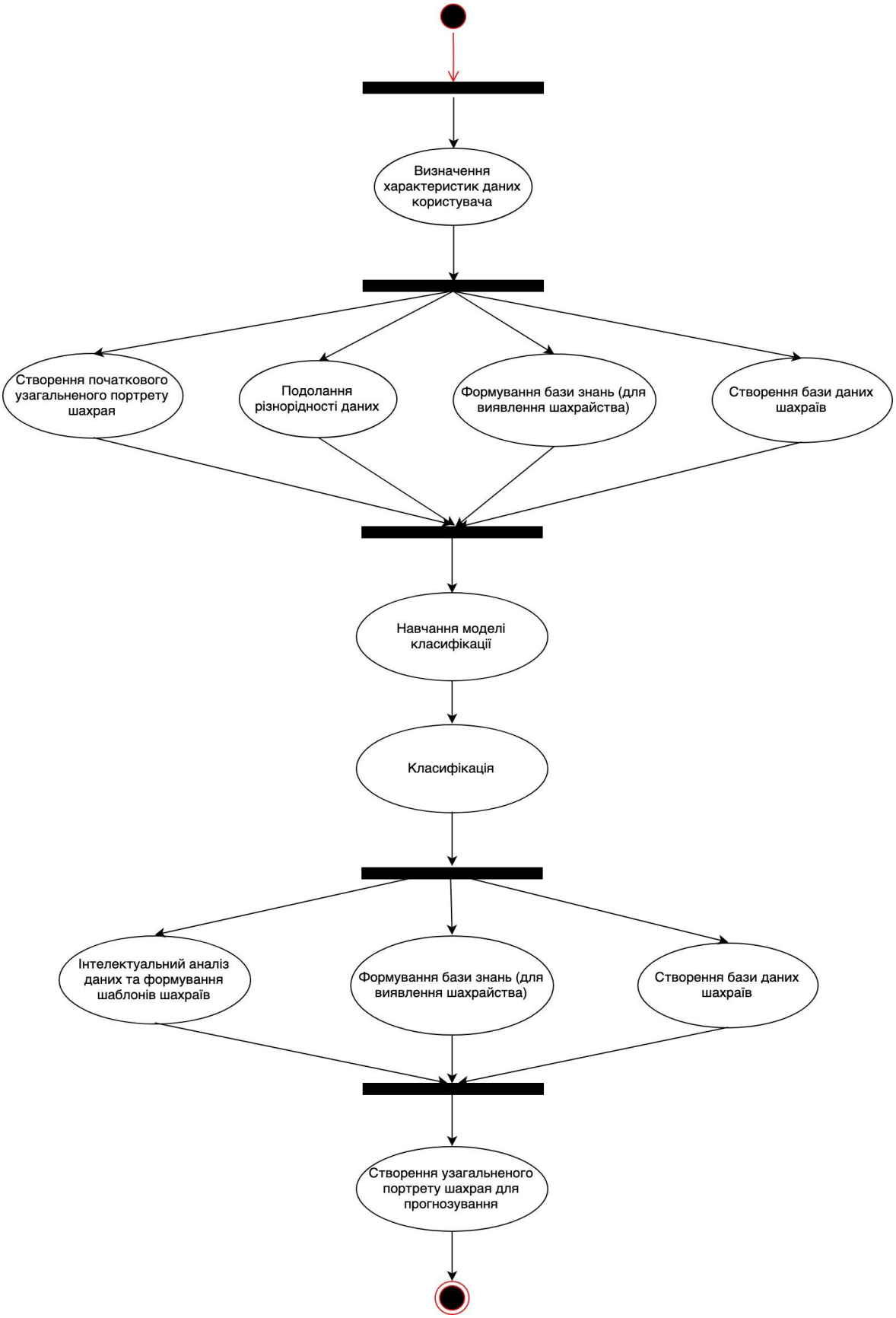


Рисунок 3.8 – UML-модель функціонування інформаційної технології виявлення шахрайства

послідовності інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків.

Здійснимо аналіз процесів в інформаційній технології виявлення шахрайства при інсталюванні мобільних додатків за допомогою діаграми послідовностіз використанням UML. На рис. 3.9 представлено діаграму послідовності інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків.

З рисунку 3.9 видно, що взаємодія відбувається між:

– користувачем інформаційної технології (органічним або шахрайським);

– адміністратором інформаційної технології;

– мобільним додатком;

– сервером;

– модулем виявлення шахрайства;

– підсистемою визначення характеристик даних користувача;

– підсистемою створення узагальненого шаблону шахрая та прогнозування;

– підсистемою подолання різнорідності даних;

– підсистемою побудови класифікаційної моделі;

– підсистемою класифікації;

– підсистемою інтелектуального аналізу даних та формування узагальненого портрету шахрая;

– базами даних;

– базами знань.

Процеси в інформаційній технології виявлення шахрайства при інсталюванні мобільних додатків відбуваються на основі запропонованих в роботі алгоритмів процесу подолання різнорідності вхідних даних, алгоритму створення узагальненого портрету шахрая, алгоритму пошуку аномалій в даних.

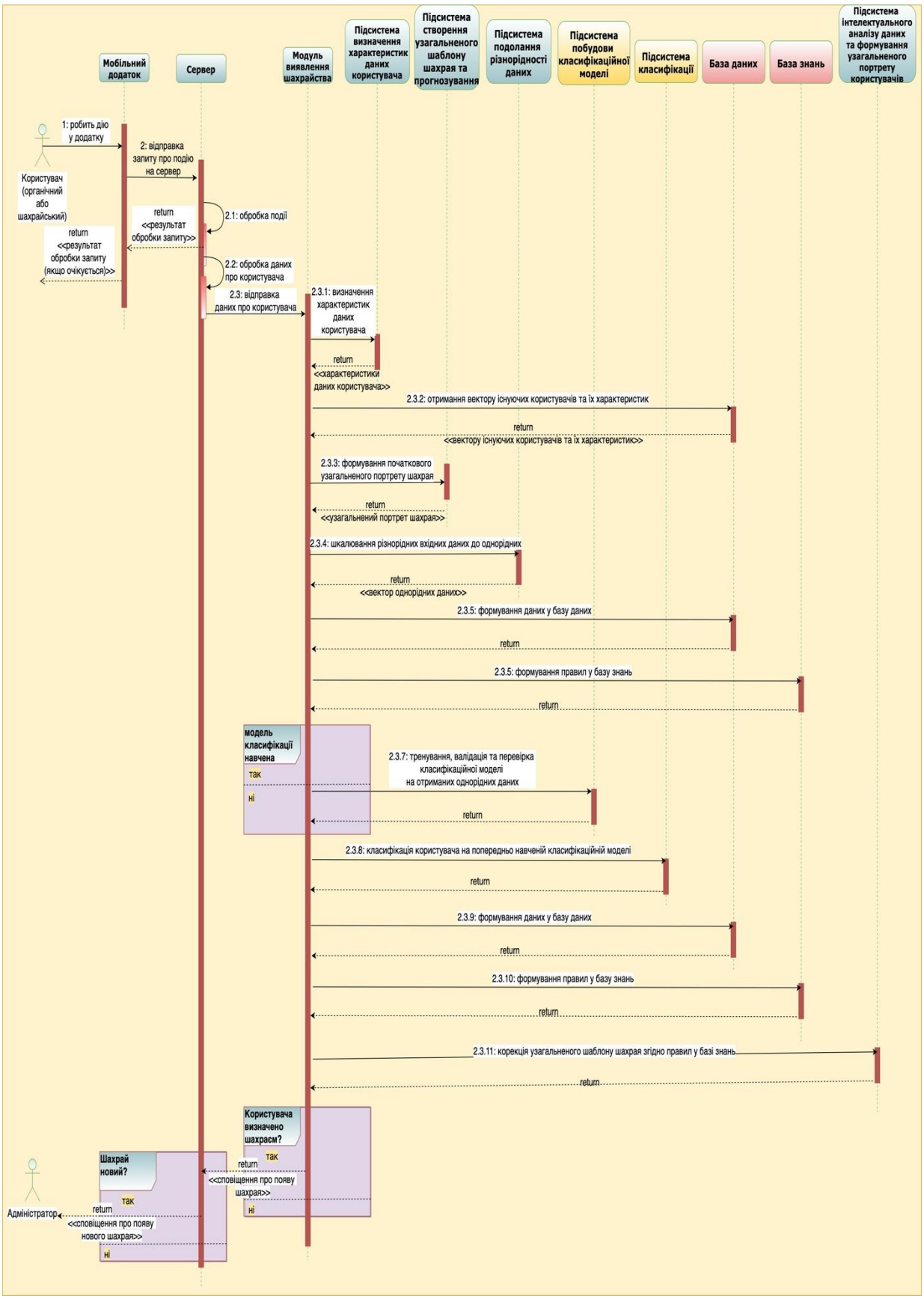


Рисунок 3.9 – Діаграма послідовності інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків

3.6 Проектування інформаційної технології виявлення шахрайства

Спроектуємо інформаційну технологію (IT) виявлення шахрайства при інсталюванні мобільних додатків, яка визначатиме шахрайських та органічних користувачів. Для повного проектування такої інформаційної технології визначимо наступне:

– вимоги та цілі інформаційної технології;

– оцінка масштабів, об’єму та обмежень інформаційної технології;

– прикладний програмний інтерфейс (API – application programming interface) інформаційної технології;

– проектування на високому рівні (high level system design);

– проектування бази даних (здійснено у пункті 3.1.2);

– оцінка об’єму інформації;

– проектування компонентів (component design);

– надійність і надлишковість інформаційної технології (reability and resundancy);

– шардинг (sharding), розбиття (partitioning) та реплікація (replication) даних;

– кешування;

– балансування навантаження (load balancing).

Розглянемо більш детально кожен із зазначених етапів проектування інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків.

Вимоги та цілі.

Запропонована інформаційна технологія зберігатиме інформацію про користувачів та їх події, а також формуватиме правила у базі знань, що розвиватиметься, з метою подальшого прогнозування та класифікації користувачів на класи «шахрай» та «органічний користувач».

Інформаційна технологія опрацьовуватиме декілька запитів:

1) запит, який повертатиме список органічних та шахрайських користувачів;

2) запит, який повертатиме список органічних та шахрайських маркетингових кампаній;

3) миттєве сповіщення по новій дії користувача про появу шахрая у випадку, якщо цей користувач буде визначений шахрайським при настанні нової події.

У зв’язку із вище визначеними запитами, які повинна опрацьовувати інформаційна технологія, тобто її цілями, визначимо вимоги, яким вона повинна відповідати.

Функціональні вимоги:

– опрацювання інформаційною технологією вище вказаних запитів та сповіщення 1-3.

Нефункціональні вимоги:

– інформаційна технологія повинна бути високодоступною (highly available). Це є необхідною вимогою, оскільки інакше при збоях інформаційної технології, вона перестане виконувати свої функції, не буде відповідати на запити і не сповіщатиме про появу шахраїв;

– сповіщення про появу шахраїв повинно відбуватися у реальному часі з мінімальною затримкою;

– затримка відповіді на запити 1-2 є допустимою, оскільки вони існують лише для інформації про минулі події і не потребує миттєвого вирішення;

– інформаційна технологія повинна бути високонадійною (highly reliable); кожна із збережених дій користувача та кожне із сформованих правил у базі знань не повинні бути втраченими;

– практично, кількість користувачів та їх дій необмежена, тому ефективне управління сховищем даних (storage) повинно бути вирішальним фактором при розробці даної інформаційної технології;

– користувачі не повинні мати доступ до правил із бази знань.

Розширені вимоги:

– аналітика; наприклад, скільки нових шахраїв з’явилося, динаміка зростання/зменшення кількості шахраїв тощо;

– розроблений сервіс повинен бути доступний через REST APIs (representational state transfer – передача репрезентативного стану) іншими сервісами з метою створення інформаційного вебсайту та мобільного додатку з системою оповіщення про появу шахраїв.

Оцінка масштабів, об’єму та обмежень інформаційної технології.

Визначивши цілі та вимоги, можна відзначити, що інформаційна технологія повинна бути готова до навантаження на читання (read-heavy), оскільки система сповіщення про появу нового користувача повинна спрацьовувати миттєво.

Оцінка необхідного об’єму.

Оцінимо необхідний об’єм, масштаб та обмеження для інформаційної технології, яку розробляємо. Нехай матимемо:

– 500 мільйонів (500M) користувачів та 1 мільйон (1M) активних користувачів щоденно;

– 15M нових подій користувачів кожного дня, тобто 173,6 нових подій користувачів щосекунди;

– середній розмір інформації по кожній події користувача – 307 байт.

Отже, об'єм, необхідний для зберігання інформації по подіям користувачів за 1 день:

$$15\ \text{M} * 307\ \text{байт} = 4{,}605\ \text{Гб}.$$

Об'єм , необхідний для зберігання інформації по подіям користувачів за 10 років:

$$4{,}605\ \text{Гб} * 365\ (\text{днів у році}) * 10\ (\text{років}) \approx 16{,}8\ \text{Тб}.$$

Більш наглядно дана інформація подана на рисунку 3.10.

Рисунок 3.10 – Оцінка об’єму, необхідного для роботи інформаційної технології на 10 років

Прикладний програмний інтерфейс (API) системи на основі інформаційної технології виявлення шахрайства.

Для виконання функціональних вимог інформаційної технології та відповідно до необхідних запитів, використовуватимемо REST API та виділимо наступний прикладний програмний інтерфейс:

1. Для запиту 1, що повертатиме список органічних та шахрайських користувачів прикладний програмний інтерфейс виглядатиме наступним чином:

getUsersByClass(startDate=None, endDate=None, campaign_id=None),

де параметрами є:

startDate (Long) – опціональне (необов’язкове поле) для пошуку користувачів, зареєстрованих після вказаної у даному параметрі дати, який задано у форматі timestamp (відмітка про час);

endDate (Long) – опціональне (необов’язкове поле) для пошуку користувачів, зареєстрованих до вказаної у даному параметрі дати, який задано у форматі timestamp (відмітка про час);

campaign_id (String) – унікальний ідентифікатор маркетингової кампанії – опціональне (необов’язкове поле) для пошуку користувачів, приведених

вказаною у даному параметрі маркетинговою кампанією. Параметр задано у форматі рядка.

Вихідні дані, які повертатимуться у відповідь на запит 1, наступні:

users (Map<String, List<UserDto>>), а саме, структура даних типу <ключ, значення>, де ключем буде тип класу («шахрай» або «органічний»), а значенням по ключу буде список з даними про користувачів відповідного класу з необхідною для відображення інформацією типу *UserDto*.

UserDto, у свою чергу, має наступні параметри:

id (String) – унікальний ідентифікатор користувача;

createdAt (long) – дата реєстрації користувача;

campaign (String) – маркетингова кампанія, що привела користувача (необов'язкове поле);

campaign_id (String) – унікальний ідентифікатор маркетингової кампанії, що привела користувача (необов'язкове поле);

fraudFeature (String) – ознака, по якій користувач був ідентифікований як шахрай (необов'зкове поле – лише для користувачів, що відносяться до класу «шахрай»).

Зауважимо, так як даний запит 1 має інформацію про шахрайські ознаки, він повинен бути доступний лише для адміністратора інформаційної технології. Безпеку та надання прав доступу буде розглянуто нижче.

На вебсайті повинна бути можливість групувати отриманих користувачів по кожному з полів класу UserDto.

Поглянемо на приклад запиту 1 та відповіді на запит 1, що матиме задані параметри та відповідь.

Приклад запиту 1:

```
https://localhost:8075/classification/getUsersByClass?startDate=150048634939&endDate=150048634939&campaign=POLHUL
```

Приклад відповіді на запит 1:


```
{
  "data": {
                "fraud": [
                    {
                         "id": "6345",
                        "createdAt": 150048634939,
                        "campaign": "POLHUL",
                        "campaign_id": "556dc524-84e9-4e33-8581-2d70a2fb4a6b",
                        "fraudFeature": "INSTALLATIONS_OF_APP_PER_DAY"
                    },
                    {
                         "id": "15573",
                        "createdAt": 150048634939,
                        "campaign": "POLHUL",
                        "campaign_id": "556dc524-84e9-4e33-8581-2d70a2fb4a6b",
                        "fraudFeature": "NOT_CONFIRMED_PURCHASE"
                    }
                ],
                "organic": [
                    {
                         "id": "9345",
                        "createdAt": 150048634939,
                        "campaign": "POLHUL",
                        "campaign_id": "556dc524-84e9-4e33-8581-2d70a2fb4a6b"
                    },
                    {
                         "id": "6368",
                        "createdAt": 150048634939,
                        "campaign": "POLHUL",
                        "campaign_id": "556dc524-84e9-4e33-8581-2d70a2fb4a6b"
                    }
                ]
  }
}
```


2. Для запиту 2, що повертатиме список органічних та шахрайських маркетингових кампаній:

getCampaignsByClass(startDate=None, endDate=None),

де параметрами є:

startDate (Long) – опціональне (необов’язкове поле) для пошуку маркетингових кампаній, які приводили користувачів після вказаної у даному параметрі дати, що задається у форматі timestamp (відмітка про час);

endDate (Long) – опціональне (необов’язкове поле) для пошуку маркетингових кампаній, які приводили користувачів до вказаної у даному параметрі дати, що задається у форматі timestamp (відмітка про час);

на вихід матимемо наступні дані:

campaigns (Map<String, List<CampaignDto>>), а саме, структура даних типу <ключ, значення>, де ключем буде тип класу («шахрай» або «органічний»),

а значенням по ключу буде список маркетингових кампаній відповідного класу з необхідною для відображення інформацією типу *CampaignDto*.

CampaignDto у свою чергу має наступні параметри:

id (String) – унікальний ідентифікатор маркетингової кампанії;

name (String) – маркетингова кампанія, що привела користувачів відповідного класу (лише «органічних» або приводила «шахрайських»);

date (long) – дата реєстрації користувача;

org_cnt (long) – кількість органічних користувачів, яких привела дана маркетингова кампанія;

fraud_cnt (long) – кількість шахрайських користувачів, яких привела дана маркетингова кампанія;

fraudFeatures (List<String>) – список шахрайських ознак, по яким користувачі від даної маркетингової кампанії були ідентифіковані як шахрайські (необов'зкове поле – лише для маркетингових кампаній, що відносяться до класу «шахрай»).

Відмітимо, що так як даний запит 2 має інформацію про шахрайські ознаки, він повинен бути доступний лише для адміністратора інформаційної технології. Безпеку та надання прав доступу буде розглянуто нижче.

На веб-сайті повинна бути можливість відсортувати або згрупувати отримані маркетингові кампанії по кожному з полів класу CampaignDto.

Поглянемо на приклад запиту 2 та відповіді на запит 2, що матиме задані параметри та відповідь.

Приклад запиту 2:

```
https://localhost:8075/classification/getCampaignsByClass?startDate=150048634939&endDate=150048634941
```

Приклад відповіді на запит 2:


```
{ "data": {
                "fraud": [
                   {
                        "id": "556dc524-84e9-4e33-8581-2d70a2fb4a6b",
                        "name": "POLHUL",
                        "date": 150048634939,
                        "org_cnt": 20,
                        "fraud_cnt": 123456,
                        "fraudFeatures": [
                           "INSTALLATIONS_OF_APP_PER_DAY",
                           "NOT_CONFIRMED_PURCHASE"
                        ]
                   },
                   {
                        "id": "93ae26f9-b8dd-4584-9efe-ce67663ffe50",
                        "name": "FRAUDULENT_ONE",
                        "date": 150048634940,
                        "org_cnt": 20000,
                        "fraud_cnt": 7450,
                        "fraudFeatures": [
                           "NOT_CONFIRMED_PURCHASE"
                        ]
                   }
                ],
                "organic": [
                   {
                        "id": "861aca5e-4b7a-45d3-86c6-adea63a9defc",
                        "name": "TANYA",
                        "date": 150048634939,
                        "org_cnt": 38654654,
                        "fraud_cnt": 0
                   },
                   {
                        "id": "3a7ff034-0eb3-472f-82ec-314e5e82157a",
                        "name": "TETIANA_S_CAMPAIGN",
                        "date": 150048634941,
                        "org_cnt": 66367,
                        "fraud_cnt": 0
                   }
                ]
  }
}
```


3. Миттєве сповіщення по новій дії користувача про появу шахрая у випадку, якщо цей користувач буде визначеним як шахрайський при настанні нової події.

Сповіщення міститиме у собі інформацію типу *UserDto* та нотифікуватиме адміністратора про появу шахрая у мобільному додатку.

Проектування на високому рівні.

На високому рівні необхідно підтримувати 2 сценарії:

– один для збору (завантаженя та зберігання) інформації про події користувачів;

– та один для пошуку користувачів та маркетингових кампаній по класам для для нотифікації при появі шахрая.

Тому необхідними є сервери баз даних для зберігання інформації про події користувачів. Також, як зазначалося у підрозділі 2.3 та пункті 3.1.4, необхідними є сервери баз знань для зберігання правил.

Високорівнева схема інформаційної технології виявлення шахрайства, яка враховує вище сказане, представлена на рис. 3.11.

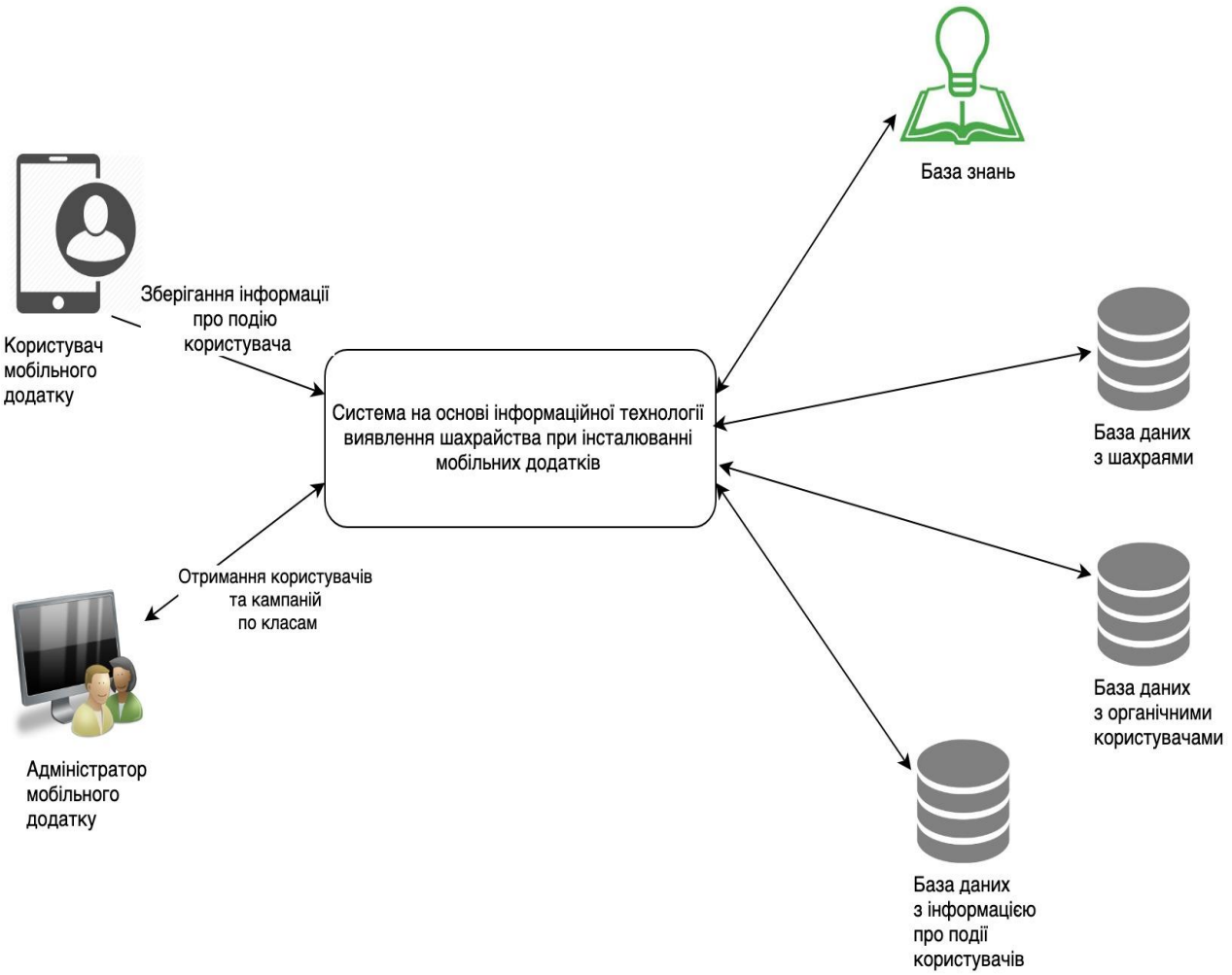


Рисунок 3.11 – Високорівнева схема інформаційної технології виявлення шахрайства

Проектування компонентів.

Необхідно зазначити, що в інформаційній технології, що розробляється, відбуватиметься як запис, так і вичитка інформації. Проте перед проектуванням інформаційної технології слід пам’ятати, що веб сервіси мають обмеження на підключення. Так наприклад, якщо веб сервер матиме ліміт у 500 підключень у будь-який момент часу, то він не зможе опрацювати більше, ніж 500 одночасних записів чи запитів (читання). Для подолання такого вузького місця, розділимо записи та читання в різні сервіси. Для цього необхідно виділити окремі сервери для запису та окремі сервери для читання (рис. 3.12). Усі компоненти представлено на рис. 2.12. Такий розділ також дозволить маштабувати та оптимізувати кожну з цих операцій (читання, запис) незалежно одна від іншої.

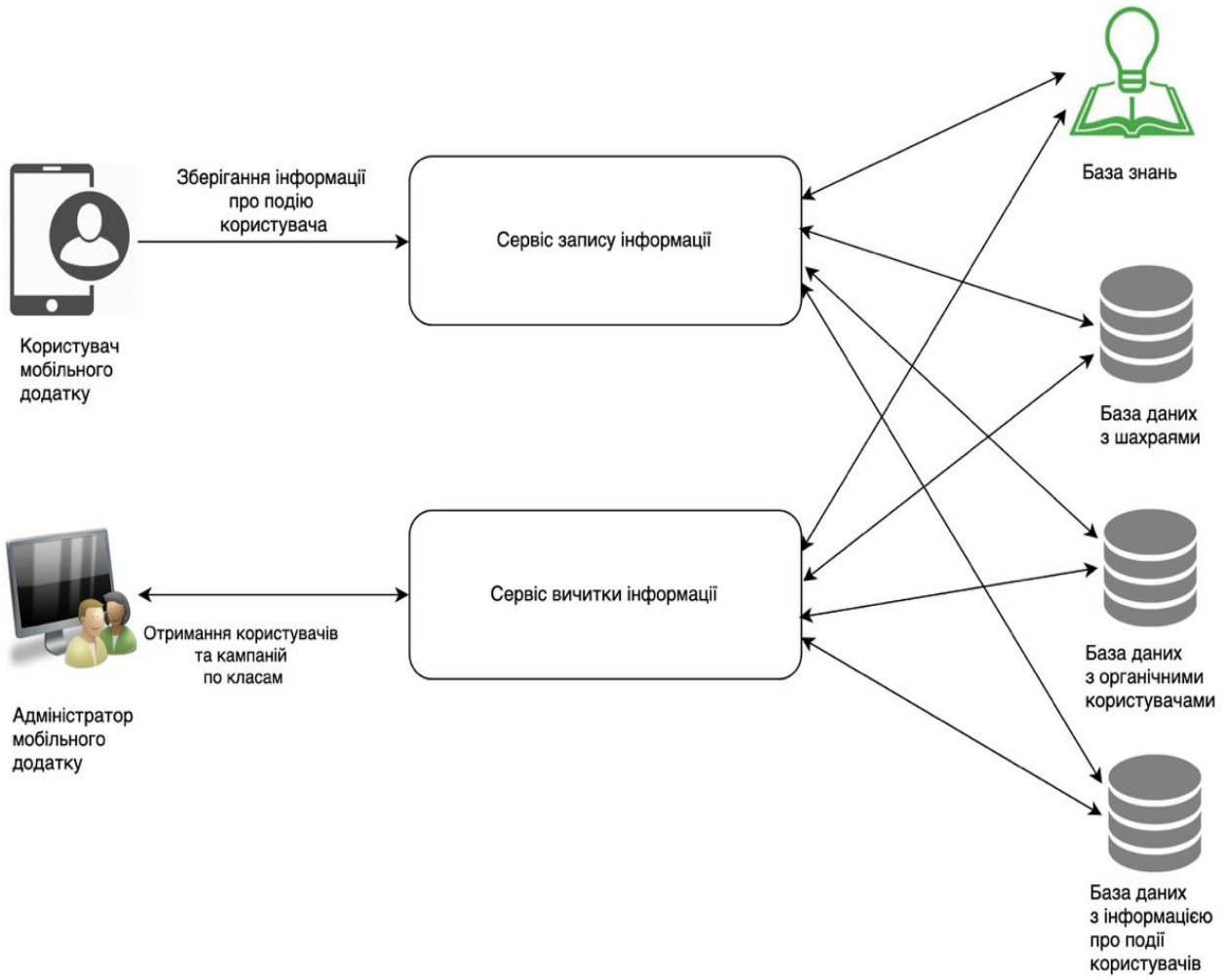


Рисунок 3.12 – Компонентне проектування інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків

Надійність і надлишковість інформаційної технології.

В інформаційній технології, що розробляється, втрачання інформації про користувачів та вже сформованих баз шахраїв та органічних користувачів і правил у базі знань не є бажаними. Тому необхідно зберігати декілька копій кожної бази. Так, якщо один із серверів зберігання даних (storage server) дасть збій / перестане відповідати / стане непрацездатним, то буде можливість отримати ці дані з іншої копії такого серверу або з іншого серверу зберігання даних.

Такий принцип також застосовується і до інших компонентів інформаційної технології. Так, для того, щоб отримати високу доступність інформаційної технології, необхідно мати кілька копій (реплік) сервісів, що працюють в інформаційній технології, так що якщо деякі сервіси дасть збій / перестане відповідати / стане непрацездатним, інформаційна технологія все ще залишатиметься доступною та запущеною. Надмірність (redundancy) видаляє єдину точку збою в інформаційній технології.

При необхідності запустити лише один примірник сервісу у будь-якій точці, можна запустити надлишкову вторинну копію (redundant secondary copy) сервісу, яка не обслуговуватиме будь-який трафік, але зможе взяти його під контроль після відмови чи проблеми з первинним сервісом.

Створення надмірності в інформаційній технології дозволяє видалити окремі точки відмови і забезпечити резервну або запасну функціональність, якщо це необхідно в умовах кризи. Наприклад, якщо є два екземпляри одного сервісу, що працюють у виробництві, та один з них виходить з ладу або погіршується, інформаційна технологія може перейти на непошкоджену копію. Така відмовостійкість (failover) може відбуватися автоматично або вимагати ручного втручання.

Схема інформаційної технології виявлення шахрайства, що є надійною та відмовостійкою та враховує усі вище вказані принципи, зображена на рисунку 3.13.

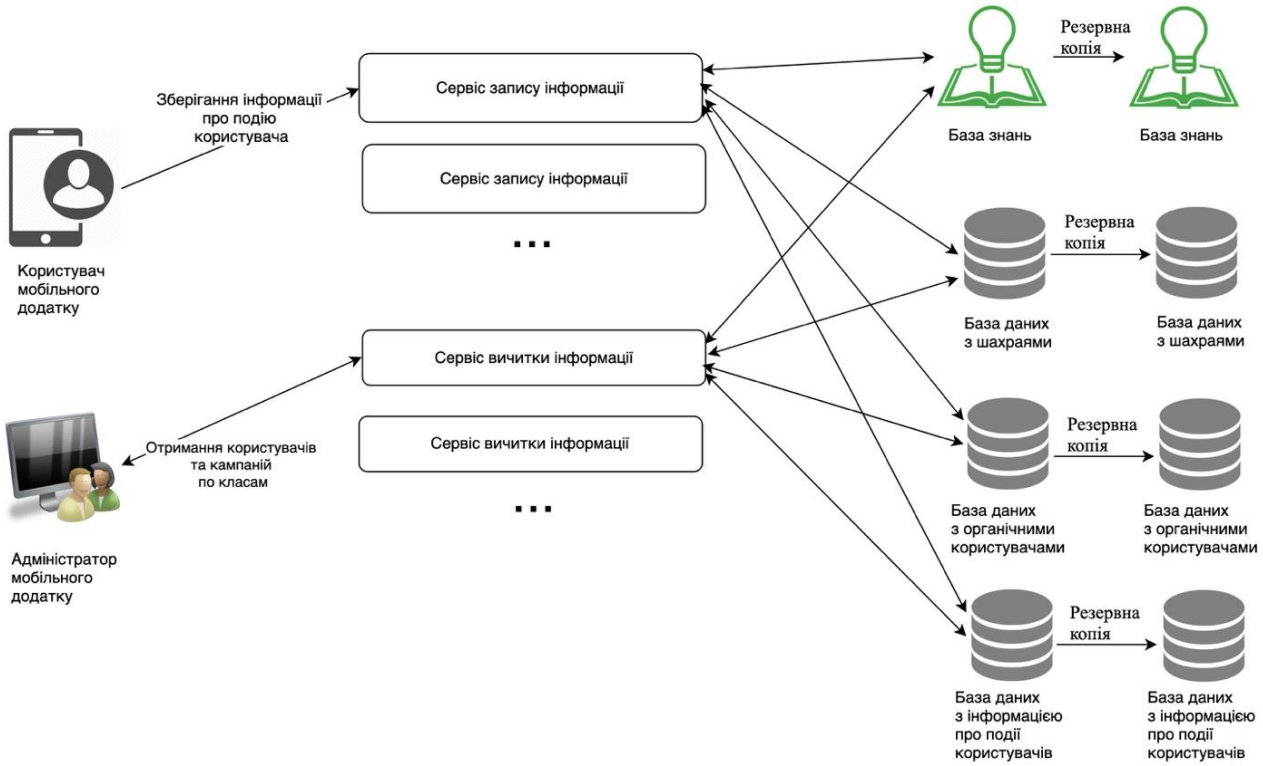


Рисунок 3.13 – Схема інформаційної технології виявлення шахрайства, що є надійною та відмовостійкою та враховує усі вище вказані принципи

Шардинг, розбиття та реплікація даних.

Шардинг (секція) – це поділ логічної бази даних або її складових елементів на окремі незалежні частини. Для того, щоб масштабувати базу даних, необхідно розділити (розбити) її так, щоб вона могла зберігати інформацію про мільярди дій користувачів. Для цього потрібно створити таку схему розділення, яка б розділяла і зберігала усі дані інформаційної технології на різних серверах баз даних (DB servers).

Розглянемо різні варіанти шардінгу для бази даних з інформацією про події користувачів:

– розділення на основі UserID. Якщо припустити, що дані розділятимуться на основі "UserID", так буде можливість зберігати всі дії користувача на одному шарді. Якщо один фрагмент бази даних (БД) – 1 ТБ, то знадобиться 17 шардів для зберігання 16.8 ТБ даних. Припустимо, що для кращої продуктивності та масштабованості в інформаційній технології виділено 20 шардів. Таким чином, ми номер шарду можна буде знайти за

UserID%20, (3.12)

та зберегти дані на шарді з отриманим номером;

– розділення на основі ActionID. Щоб однозначно визначити будь-яку подію в інформаційній технології, можна додати номер шарда з кожним ActionID. Проте у такому розділенні немає необхідності.

Кешування.

Для того, щоб адміністратор мав швидкий доступ до останніх визначених шахраїв або ж маркетингових кампаній, що привели цих шахраїв, можна ввести кеш перед базою даних. Для цього можна використовувати готові рішення, такі як Memcache, які можуть зберігати всі дані про останніх визначених шахраїв та їх маркетингові компанії. Сервери веб-сторінки, перш ніж зробити запит в базу даних, можуть швидко перевірити наявність даних в кеші.

Визначимо, скільки ж кешу необхідно мати? Доцільно почати з 20% щоденного трафіку, і, виходячи з шаблону використання користувачів (clients' usage pattern), можна буде налаштувати, скільки серверів для кешу потрібно виділити для продуктивної роботи інформаційної технології. За оцінкою вище, потрібно 0,921 ГБ пам'яті для кешування 20% щоденного трафіку. Оскільки сучасний сервер може мати 256 Гб пам'яті, то можна легко помістити весь кеш в одну машину. Крім того, можна використовувати кілька менших серверів для зберігання останніх визначених шахраїв або ж шахрайських маркетингових кампаній.

Яка ж політика вилучення кеша найкраще відповідає потребам інформаційної технології, що розробляється? Для цього спершу необхідно відповісти на наступне запитання: коли буде заповнений кеш, і з'явиться необхідність замінити шахрая на новішого, яку політику вилучення кеша було б доцільно вибрати у цьому випадку? Розумною політикою для такого випадку буде черга – перший прийшов перший пішов (First In First Out – FIFO). Згідно з

цією політикою, спочатку видаляємо «найстарішого» шахрая, який знаходиться у кеші, не зважаючи на те, як часто він зустрічається. Для зберігання шахраїв і хешу, можна використовувати Linked Hash Map (двозв'язну карту – структуру даних типу <ключ, значення>) або подібну структуру даних, яка також відстежуватиме шахраїв, яких нещодавно діставали з бази даних.

Щоб ще більше підвищити ефективність, можна скопіювати сервери кешування, щоб розподілити навантаження між ними.

Балансування навантаження.

В інформаційній технології, що розробляється, доцільно додати шари балансувальника навантаження у двох місцях:

– між клієнтами і серверами додатків;

– між серверами додатків і сервером з логікою;

Для початку може бути прийнятий простий циклічний підхід («round robin»), який буде поширювати всі вхідні запити однаково серед серверів з логікою. Цей LB є простим у реалізації і не вводить ніяких накладних витрат. Ще одна перевага цього підходу полягає в тому, що сервер якщо недоступний, то балансувальник навантаження виведе його зі списку серверів, на які можна давати навантаження і перестане надсилати до нього будь-який трафік.

Проблема з round robin LB у тому, що не враховується завантаження сервера. Якщо сервер перевантажений або уповільнений, балансування навантаження не припинить надсилання нових запитів до цього сервера. Щоб вирішити це, потрібне більш інтелектуальне рішення LB, яке періодично надсилатиме запит до сервера з логікою про його навантаження і регулюватиме трафік на основі цього.

3.7 Висновки до розділу 3

Розроблено інформаційну технологію з використанням системного аналізу та моделювання, а саме: побудована логічна, концептуальна моделі

інформаційної технології. Здійснено аналіз та розробку структур даних інформаційної технології, розроблено шаблон шахрая для формування портрету шахрая. Як показали дослідження, усі розроблені моделі є необхідними для ефективного функціонування інформаційної технології виявлення шахрайства на основі запропонованого в роботі узагальненого методу, в основі якого лежать моделі, методи та алгоритми виявлення аномалій в даних.

Розроблено алгоритми виявлення шахрая при інсталюванні мобільних додатків та алгоритм створення узагальненого портрету шахрая на основі методів, моделей та алгоритмів, розроблених у 2-му розділі, які є основою запропонованої в роботі інформаційної технології.

Розроблено алгоритм пошуку аномалій в даних як основа функціонування інформаційної технології за допомогою діаграми activity, а також здійснено аналіз процесів в інформаційній технології виявлення шахрайства за допомогою UML-діаграми послідовності.

Запропоновано інформаційну технологію виявлення шахрайства при інсталюванні мобільних додатків, яка використовує запропоновані метод виявлення шахрайства при інсталюванні мобільних додатків, метод подолання різнорідності вхідних даних, модель класифікації даних, і, на відміну від існуючих систем Antifraud, дозволила підвищити точність класифікації користувачів до 99,14 %, зокрема точність класифікації шахраїв – до 82,76 %.

Результати проведених досліджень опубліковані в [16], [18], [24], [26]-[28], [61], а також представлено у University of Cambridge (Додаток Д).

РОЗДІЛ 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВИЯВЛЕННЯ ШАХРАЙСТВА ПРИ ІНСТАЛЮВАННІ МОБІЛЬНИХ ДОДАТКІВ З ВИКОРИСТАННЯМ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

Результати моделювання та розроблені методи, моделі та алгоритми, отримані в попередніх розділах, є основою для розробки інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків.

Інформаційна технологія реалізована у вигляді пакета програм і даних програмного забезпечення «Mobile app install fraud detection system», які дозволяють здійснювати класифікацію користувачів на органічних та шахрайських на основі розроблених моделі, методів та алгоритмів.

4.1 Аналіз та підготовка вхідних даних для проведення експериментальних досліджень з використанням інформаційної технології виявлення шахрайства

Під час вибору набору даних існувала проблема відсутності загальновідомих відкритих наборів даних про діяльність користувача в мобільних додатках. А користувачі мобільного додатку, в розробці яких брав участь автор статті, не були помічені на класи "органічний користувач" або "шахрай", що ускладнило б процес оцінки точності. Тому помічені набори даних були отримані від вищезгаданих експертів (Додаток В), які мали досвід роботи в даній предметній області. Також, дослідження проводились на вибірці з ресурсу Kaggle [95].

Розглянемо та проаналізуємо другу вибірку даних, перевіримо чи є вона репрезентативною та скільки в ній унікальних записів. Оцінка наданого набору даних здійснювалась з використанням мови програмування Python та бібліотек numpy, pandas, matplotlib. Поглянемо на перші 5 рядків набору даних на рис. 4.1.

V5	V6	V7	V8	V9	...	V21	V22	V23	
38321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066
60018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339
03198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689
10309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175
07193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141

а)

```
appActions = pd.read_csv('app-actions.csv')
appActions.head()
```

V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

б)

Рисунок 4.1 – Приклад вхідних даних з наданого набору даних:
а – частина 1, б – частина 2

Кожен рядок набору даних – це окремий запис про дію користувача під час та після інсталювання мобільного додатку. Кожен запис представлений декількома характеристиками (features), які наведені у колонках таблиці. Дані, які містяться у файлі app-actions.csv, презентують список дій користувачів.

Розглянемо значення кожної колонки:

– *Time* – кількість секунд між поточною дією та першою дією в наборі даних;

– *V1, V2, …, V28* – ознаки (features) кожної з інсталяцій, які представляють собою дії користувачів тощо;

– *Class* – 1 – це шахрайська інсталяція, 0 – органічна інсталяція.

У наборі даних представлено 284807 користувачів та 31 характеристика (features) (рис. 4.2).

```
(28481, 31)
```

Рисунок 4.2 – Кількість користувачів та кількість різних характеристик у наданому наборі даних

Маємо 2 типи класів-міток у наборі даних – органічні та шахрайські (рис. 4.3).

```
print(appActions['Class'].unique())
[0 1]
```

Рисунок 4.3 – Визначення кількості класів у наданому поміченому наборі даних

Подивимося розподіл користувачів по класам у наборі даних (рис. 4.4-4.5).

```
print(appActions.groupby('Class').size())
Class
0    284315
1       492
dtype: int64
```

Рисунок 4.4 – Розподіл користувачів по наявним класам у наданому наборі даних

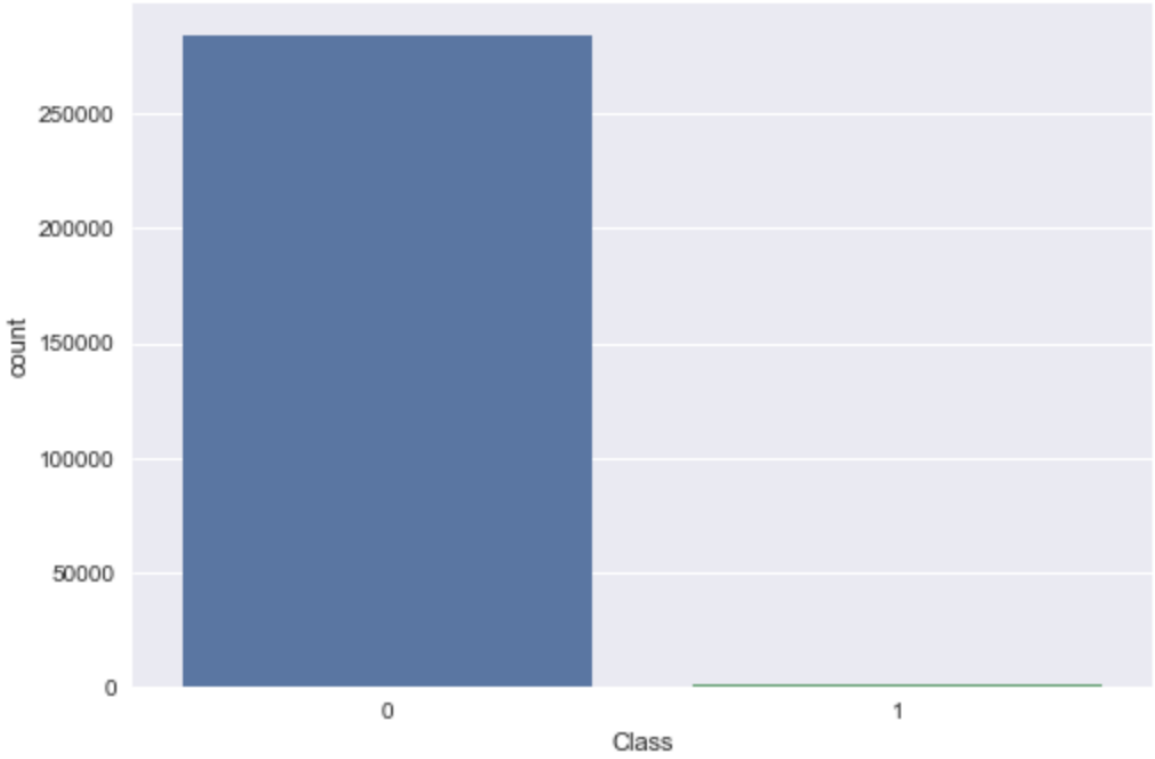


Рисунок 4.5 – Розподіл користувачів по наявним класам у наданому наборі даних

З рис. 4.4 – 4.5 видно, що у тестовому наборі даних є 284315 органічних інсталяцій (користувачів) та 492 шахрайських інсталяцій (користувачів).

Подивимося статистичний звіт по наданому набору даних (рис. 4.6 – частина 1, 2).

50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	'

8 rows × 31 columns

а)

	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
count	70e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	284807.000000	284807.000000
mean	62e-16	-3.444850e-16	2.578648e-16	4.471968e-15	5.340915e-16	1.687098e-15	-3.666453e-16	-1.220404e-16	88.349619	0.001727
std	40e-01	7.257016e-01	6.244603e-01	6.056471e-01	5.212781e-01	4.822270e-01	4.036325e-01	3.300833e-01	250.120109	0.041527
min	038e+01	-1.093314e+01	-4.480774e+01	-2.836627e+00	-1.029540e+01	-2.604551e+00	-2.256568e+01	-1.543008e+01	0.000000	0.000000
25%	949e-01	-5.423504e-01	-1.618463e-01	-3.545861e-01	-3.171451e-01	-3.269839e-01	-7.083953e-02	-5.295979e-02	5.600000	0.000000
50%	017e-02	6.781943e-03	-1.119293e-02	4.097606e-02	1.659350e-02	-5.213911e-02	1.342146e-03	1.124383e-02	22.000000	0.000000
75%	72e-01	5.285536e-01	1.476421e-01	4.395266e-01	3.507156e-01	2.409522e-01	9.104512e-02	7.827995e-02	77.165000	0.000000
max	84e+01	1.050309e+01	2.252841e+01	4.584549e+00	7.519589e+00	3.517346e+00	3.161220e+01	3.384781e+01	25691.160000	1.000000

б)

Рисунок 4.6 – Статистичний звіт по наданому набору даних: а – частина 1, б – частина 2

Для більшого розуміння даних, візуалізуємо їх. На рис. 4.7 зображено гістограми кожної чисельної вхідної характеристики.

4.2 Розробка методики проведення експериментальних досліджень шахрайства при інсталюванні мобільних додатків

Для того, щоб перевірити ефективність розробленого методу виявлення шахрайства при інсталюванні мобільних додатків, оцінимо точність отриманих результатів. Для проведення дослідження представленої інформаційної технології було розроблено програмне забезпечення [92]-[94], в основі якого використовуються запропоновані у роботі модель, метод та алгоритми виявлення аномалій. Дослідження проводилося в наступному порядку (алгоритм 4.1):

1. Отримання даних з бази даних користувачів та бази знань.
2. Виявлення характеристик користувача.
3. Ініціалізація узагальненого портрету шахрая.
4. Розбиття даних на дві групи $G\bar{1}$ та $\bar{G}2$ згідно правил, описаних у процедурі виявлення аномалій у різнорідних даних (2.26).

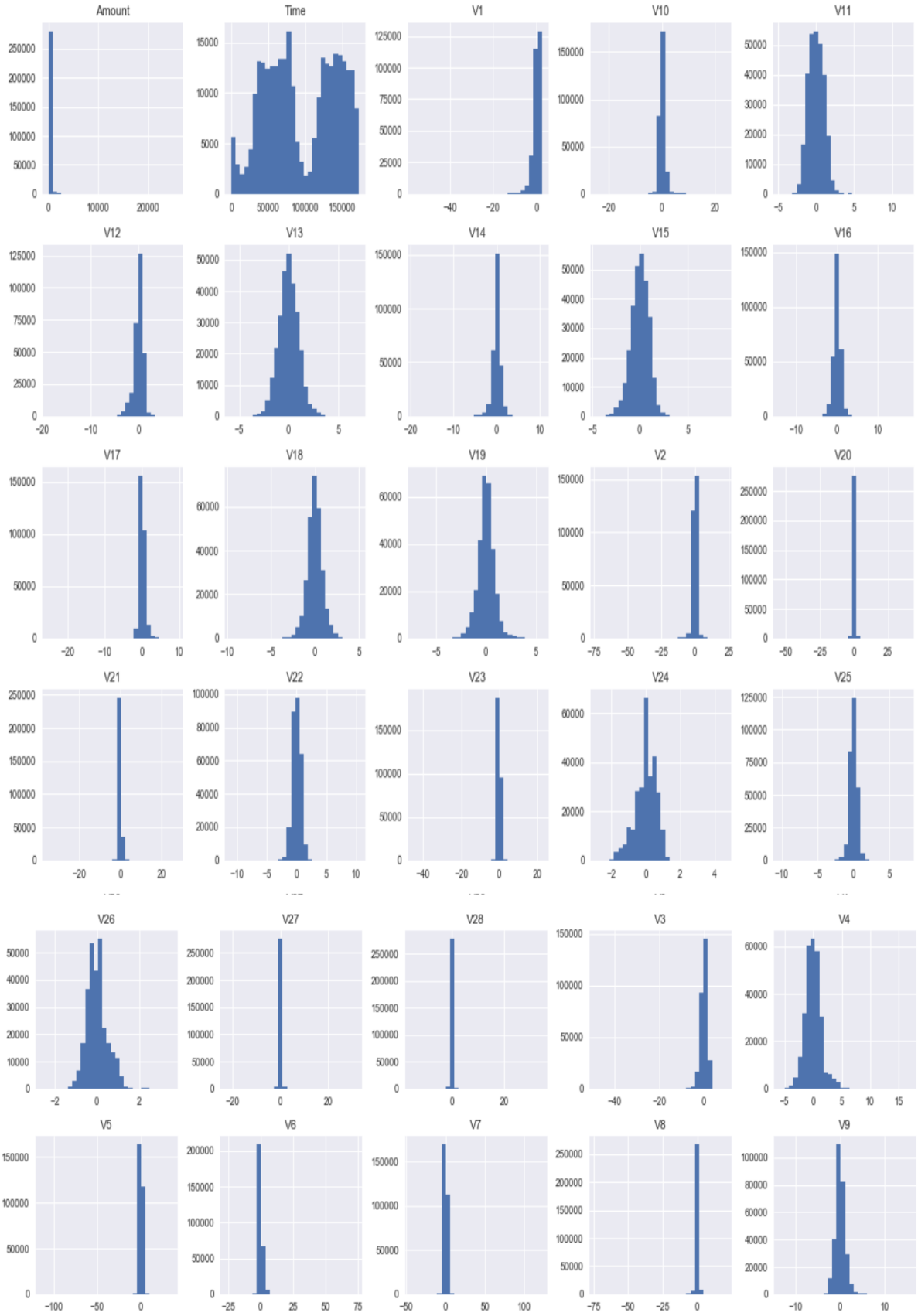


Рисунок 4.7 – Гістограми кожної чисельної вхідної характеристики з наданого набору даних

5. Метод подолання різнорідності вхідних даних та виявлення спільних характеристик по усім користувачам:

5.1 Визначення однозначних користувачів та запис їх у базу знань за формулами 2.2 – 2.14 та 3.1 – 3.8.

5.1.1 Визначення однозначних шахраїв та запис їх у базу знань.

5.1.2 Визначення однозначних органічних користувачів та запис їх у базу знань.

5.1.3 Визначення підозрілих користувачів та запис їх у базу знань.

5.2 Виявлення ознак користувачів за формулами 2.2 – 2.14.

5.2.1 Виявлення ознак шахраїв.

5.2.2 Виявлення ознак органічних користувачів.

5.2.3 Виявлення ознак підозрілих користувачів.

5.3 Виявлення коефіцієнтів схожості користувача, використовуючи характеристики з групи $\overline{G}2$ та запис їх у базу знань за формулами 3.1 – 3.8 та згідно [13], [45], [63], [96].

5.3.1 Виявлення коефіцієнтів схожості користувачів до шахраїв за різними ознаками.

5.3.2 Виявлення коефіцієнтів схожості користувачів до органічних за різними ознаками.

5.3.3 Виявлення коефіцієнтів схожості користувачів до підозрілих за різними ознаками.

5.3 Об'єднання конфіцієнтів схожості користувачів за різними ознаками за формулою 3.9.

6. Навчання моделі класифікації, маючи вже помічені дані завдяки попередньо отриманим однозначним користувачам з використанням формули 3.10.

7. Класифікація користувачів з використанням формули 3.10.

8. Визначення шахраїв, органічних користувачів та органічних, проте підозрілих.

9. Формування портрету шахрая за формулами 3.1 – 3.8.

Схему експериментального дослідження виявлення шахрайства як аномалій в різнорідних даних при інсталюванні мобільних додатків, в основі якої використовується представлений алгоритм, подано на рис. 4.8.

Для доведення адекватності розробленої моделі класифікації, що прийматиме на вхід однорідні дані, отримані після використання методу подолання різнорідності даних, необхідно правильно здійснити її оцінку. Для цього визначимо, що насамперед модель повинна бути наділена прогностичною здатністю та добре узагальнюватися на нових даних. Також, необхідно зауважити, що для коректного навчання та оцінки моделі, а також для уникнення перенавчання моделі, необхідно мати три набори даних: тренувальний, перевірочний (валідаційний) та контрольний. Оскільки некоректно навчати та тестувати модель на одному і тому ж наборі даних – у такому випадку результати будуть хибні. Тому модель буде тренуватися на тренувальному наборі даних, оцінюватися – на валідаційному, а для створення кінцевої версії моделі буде тестуватися на контрольному наборі даних.

Застосуємо запропоновану методику проведення експериментальних досліджень на практиці та проаналізуємо отримані результати.

4.3 Розробка алгоритму мінімізації часу виявлення шахраїв на основі розпаралелення обчислювальних процесів

Особливістю запропонованих в підпункті 2.3.1.1 коефіцієнтів, які дозволяють виконати перехід від різнорідних до однорідних даних, є те, що їх обробка йде паралельно, а це пришвидшує час обробки.

Виходячи з цього, в даному підрозділі розглянута декомпозиція задач для найбільш ресурсоємних методів та алгоритмів, запропонованих у розділі 2 і необхідних для розв’язання задачі оптимізації часу виявлення шахраїв. Будемо використовувати граф залежності задач (task dependency graph) [102] як засіб моделювання розпаралелення.

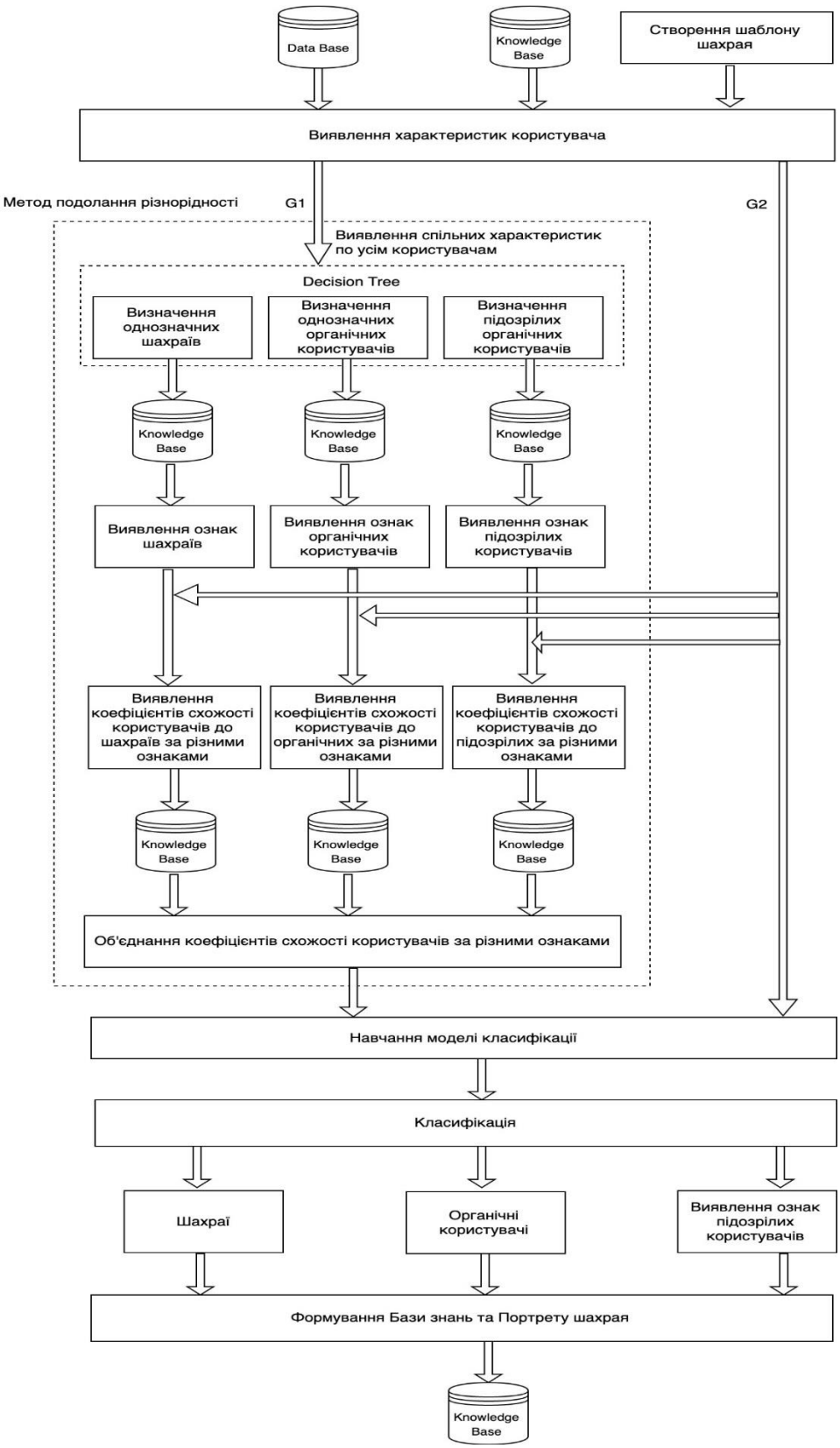


Рисунок 4.8 – Схема експериментального дослідження виявлення шахрайства, як аномалій в різнорідних даних, при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних

Для оцінювання прискорення обчислювального процесу за рахунок розбиття алгоритмів на незалежні блоки і виконання їх незалежними агентами, в даній роботі пропонується використання закону Амдала [103], [104]. Якщо не враховувати часові витрати на комунікацію між потоками, в яких виконуватимуться процеси, прискорення обчислюється як

$$S = \frac{1}{\alpha + \frac{1-\alpha}{p}},$$

де α – частка обчислень, що виконуються послідовно,

$1-\alpha$ – частка обчислень, яка може бути розпаралелена ідеально,

p – кількість задіяних вузлів.

Будемо використовувати наступний алгоритм мінімізації часу виявлення шахраїв на основі розпаралелення обчислювальних процесів:

1. Нехай задано вектор наявних різнорідних вхідних даних по користувачу $\vec{M}$ розміром m, як описано у підпункті 2.3.1.1.

2. Для побудови перетвореного вектору однорідних даних $\vec{O}$, вектор різнорідних даних розбивається на N частин таким чином, що кожний доступний потік визначає значення кожного з N коефіцієнтів згідно до запропонованого в роботі алгоритму 1 подолання різнорідності вхідних даних при інсталюванні мобільних додатків, описаного в підпункті 2.3.1.2.

3. Визначимо незалежні операції для алгоритму 1 подолання різнорідності вхідних даних при інсталюванні мобільних додатків. Загальна кількість компонентів системи $N+1$. Нехай

A_i – операція визначення значення одного з N коефіцієнтів;

B – операція агрегування даних.

4. Граф залежності задач для подолання різнорідності вхідних даних при інсталюванні мобільних додатків наведений на рис. 4.9.

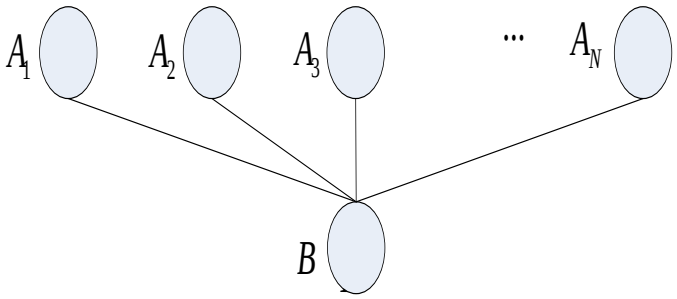


Рисунок 4.9 – Граф залежності задач для алгоритму 1 подолання різнорідності вхідних даних при інсталюванні мобільних додатків

5. Частка $\alpha = 1/(N+1)$ від загального обсягу обчислень може бути отримана лише послідовними розрахунками, а, відповідно, $1-\alpha$ може бути розпаралелена ідеально. Кількість обчислювальних систем, на яких виконується процес, $p = N+1$. Тоді прискорення від застосування розпаралеленого підходу становить

$$S = \frac{1}{1/(N+1) + \dfrac{(1-1/(N+1))}{N+1}}.$$

В результаті експериментальних досліджень було отримано підвищення швидкодії на 9,48 %.

4.4 Дослідження адекватності моделей, точності та швидкодії методу виявлення шахрайства, аналіз результатів тестування

Зазначимо, що адекватна модель повинна бути наділена двома властивостями [33]: вона наділена прогностичною здатністю та добре узагальнюється для даних, які їй ще не зустрічалися. Для оцінки цих властивостей визначається метрика похибок (наскільки сильно помиляється модель) та стратегія перевірки адекватності. Найпоширенішою метрикою похибок, що підходить для запропонованої моделі, є метрика – частота помилок класифікації. Вона полягає у тому, що при побудові моделей [97], в основі яких

лежить інтелектуальний аналіз даних, важливо вміти їх оцінювати для вибору найкращих для вирішення поставленої задачі. Розглянемо оцінки похибок моделей, які вирішують завдання класифікації і апроксимації залежностей, для оцінки запропонованої моделі. Ефективність моделі чи методу, призначених для вирішення задачі класифікації, зазвичай оцінюється за допомогою коефіцієнта помилки (error rate). Якщо класифікатор правильно визначає клас спостереження, то має місце успіх, в іншому випадку – помилка. Коефіцієнт помилки – це кількість помилок, допущених на всій множині у відношенні до загальної кількості спостережень. Він показує загальну ефективність моделі класифікації (класифікатора). Помилки класифікатора повинні визначатися на даних, на яких класифікатор не навчався. Таким чином визначається помилка узагальнення.

Отже, реалізуємо запропонований в підрозділі 2.3 метод виявлення шахрайства при інсталюванні мобільних додатків з використанням мови програмування Python. Проте, для початку зазначимо, що для отримання адекватних результатів не можна тестувати та навчати модель на одному і тому ж наборі. Також слід зазначити, що нашою метою є створення узагальненої моделі виявлення шахрайства при інсталюванні мобільних додатків, яка даватиме якісний прогноз на даних, що не брали участь у навчанні. Перенавчання ж являється основною перешкодою в досягненні даної мети. Тому для адекватної оцінки моделі та уникнення перенавчання поділимо доступний набір даних на три набори [34], [36]: тренувальний, перевірочний (валідаційний) та тестовий (контрольний). Модель навчатиметься на тренувальних даних та оцінюватиметься на перевірочних, а після створення кінцевої версії моделі – тестуватиметься на контрольних даних.

Здійснивши експерементальні дослідження за схемою експериментального дослідження виявлення шахрайства як аномалій в різнорідних даних при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних, представленою на рис. 4.12, в основі якої

використовується алгоритм 4.1, в результаті проведення пунктів алгоритму 1 – 4 отримано множину однорідних даних, приклад яких представлено на рис. 4.10.

Виконавши наступні кроки алгоритму 4.1, сформуємо графіки втрат (рис. 4.11) та точності (рис. 4.12) на етапах навчання та перевірки класифікаційної моделі, на вхід якої було подано отримані однорідні дані.

```
appActions.head()
```

Out[2]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...
0	0.0	0.359807	0.072781	0.536347	0.378155	0.338321	0.462388	0.239599	0.098698	0.363787	...
1	0.0	0.191857	0.266151	0.166480	0.448154	0.060018	0.082361	0.078803	0.085102	0.255425	...
2	1.0	0.358354	0.340163	0.773209	0.379780	0.503198	0.800499	0.791461	0.247676	0.514654	...
3	1.0	0.966272	0.185226	0.792993	0.863291	0.010309	0.247203	0.237609	0.377436	0.387024	...
4	2.0	0.158233	0.877737	0.548718	0.403034	0.407193	0.095921	0.592941	0.270533	0.817739	...

Рисунок 4.10 – Однорідні дані по кожному користувачу, отримані в результаті використання методу подолання різнорідності даних

Зазначимо, що навчання моделі класифікації проводилось у декілька епох. На рис. 4.11 та 4.12 наглядно видно, що в процесі навчання точність класифікаційної моделі зростає, а втрати зменшуються.

Для уникнення перенавчання здійснюємо навчання моделі у 100 епох, а потім оцінюємо отриманий результат на контрольних даних. Результати експерименту представлено на рис. 4.13. Точність методу виявлення шахрайства, який працює з даними, отриманими з методу подолання різнорідності вхідних даних, рівна 99,14%. Наявних шахраїв метод визначає з точністю 82,76%, загалом точність визначення класу користувачів – 99,14%, як вказано на рис. 4.14.

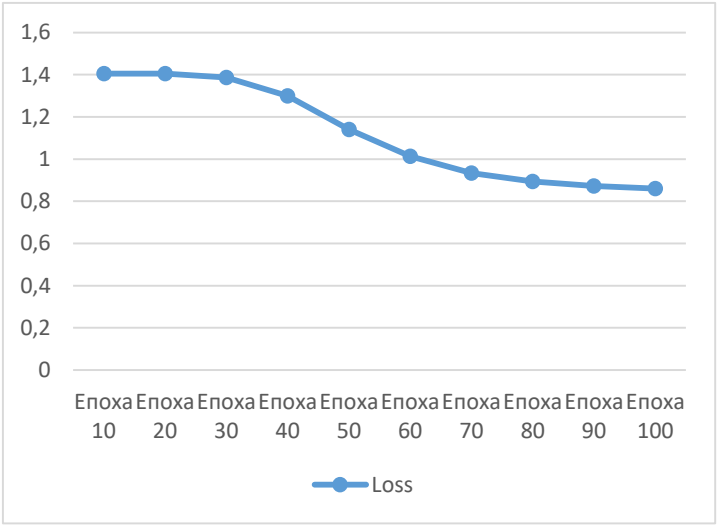


Рисунок 4.11 – Втрати на етапах навчання та перевірки

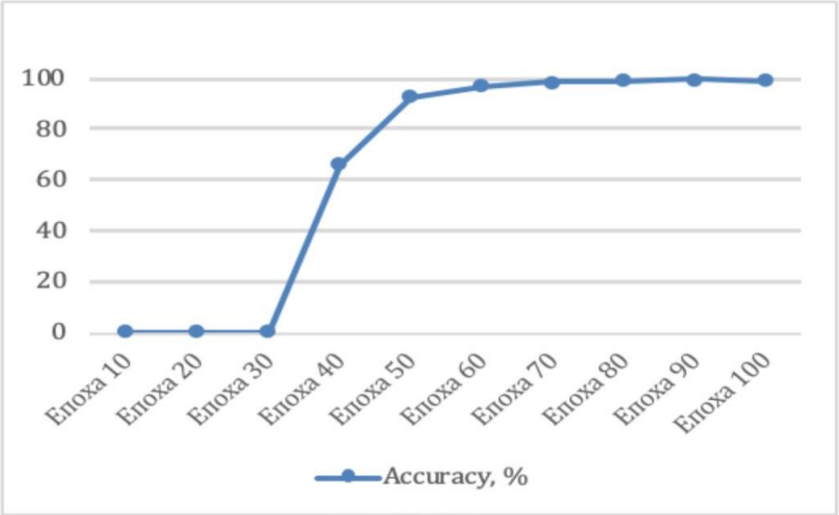


Рисунок 4.12 – Точність на етапах навчання та перевірки

```
Percent of fraudulent transactions:  0.0017274856306200 34
/Users/tetianapolhul/PycharmProjects/CreditCardFraudDetecti
  return f(*args, **kwds)
2018-07-24 23:33:11.755811: I tensorflow/core/platform/cpu_
Epoch: 0 Current loss: 1.4053 Elapsed time: 1.58 seconds
Current accuracy: 0.15%
Epoch: 10 Current loss: 1.4053 Elapsed time: 1.32 seconds
Current accuracy: 0.15%
Epoch: 20 Current loss: 1.3875 Elapsed time: 1.20 seconds
Current accuracy: 0.15%
Epoch: 30 Current loss: 1.3002 Elapsed time: 1.25 seconds
Current accuracy: 66.30%
Epoch: 40 Current loss: 1.1396 Elapsed time: 1.21 seconds
Current accuracy: 93.02%
Epoch: 50 Current loss: 1.0138 Elapsed time: 1.21 seconds
Current accuracy: 97.49%
Epoch: 60 Current loss: 0.9332 Elapsed time: 1.29 seconds
Current accuracy: 99.00%
Epoch: 70 Current loss: 0.8944 Elapsed time: 1.14 seconds
Current accuracy: 99.46%
Epoch: 80 Current loss: 0.8729 Elapsed time: 1.33 seconds
Current accuracy: 99.65%
Epoch: 90 Current loss: 0.8608 Elapsed time: 1.16 seconds
Current accuracy: 99.62%
Final accuracy: 99.14%
Final fraud specific accuracy: 82.76%

Process finished with exit code 0
```

Рисунок 4.13 – Результати комп’ютерного моделювання, здійсненого на основі даних, отриманих з методу подолання різнорідності вхідних даних

Отже, отримавши адекватну модель, побудуємо матрицю невідповідності (confusion matrix) (рис. 4.14) для порівняння отриманих результатів з очікуваними. Для перевірки моделі взято контрольну вибірку з 56962 записами, 56657 з яких – відповідають органічним користувачам, а 305 – користувачам-шахраям.

Рисунок 4.14 – Матриця невідповідності контрольної вибірки без нормалізації

Можна подати у такому вигляді, як показано на рис. 4.15.

Істинно позитивна (ІП) = 56219	**Хибно позитивна** (ХП) = 52
Хибно негативна (ХН) = 438	**Істинно негативна** (ІН) = 253

Рисунок 4.15 – Матриця невідповідності контрольної вибірки без нормалізації з визначенням кожного значеня для подальших обрахунків

З матриці невідповідностей, поданої на рис. 4.14 та 4.15, видно, що 56219 об'єктів першого класу («органічний користувач»), що складає 99,22% від усіх 56657 об'єктів, віднесених до першого класу. 438 об'єкт першого класу помилково класифіковані як об'єкти другого класу («шахрай»). Правильно класифіковано 99,22% об'єктів першого класу, а 0,78% класифіковані неправильно. Щодо об’єктів другого класу – користувачів-шахраїв – 253 об’єкти

з 305, віднесених до другого класу, класифіковано правильно, а 52 – помилково віднесено до першого класу. Таким чином, правильно класифіковано 82,95% користувачів-шахраїв, а 17,05% класифіковано неправильно. В цілому правильно розцінено 99,14% об'єктів (56219 + 253 = 56472 об'єктів з 56962 об'єктів), неправильно класифіковано 0,86% об'єкти. Нормалізована матриця невідповідності для контрольної вибірки представлена на рис. 4.16. Зазначимо, що оскільки 99,14% правильно визначених об’єктів загалом та 82,95% правильно визначених користувачів-шахраїв при моделюванні – це висока точність, тому впровадження, автоматизація даної моделі та побудова запропонованої інформаційної технології з використанням моделі є доцільними.

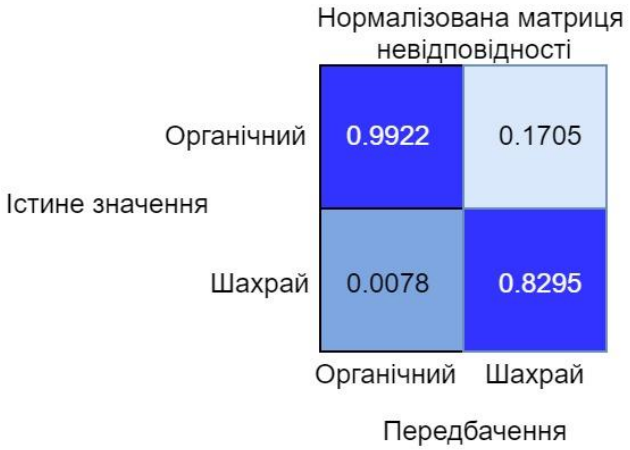


Рисунок 4.16 – Нормалізована матриця невідповідності контрольної вибірки

Слід зазначити, що у бінарній класифікації кожне окреме передбачення може мати чотири результати [97]:

– істинно позитивний – ІП (True Positive) TP – користувач є шахрайським, результат класифікації позитивний;

– істинно негативний – ІН (True Negative) TN – користувач органічний, результат класифікації негативний;

– хибно позитивний – ХП (False Positive) FP – користувач органічний, результат класифікації позитивний – також називається помилкою I роду або помилковою тривогою (викликом);

– хибно негативний – ХН (False Negative) FN – користувач є шахрайським, результат класифікації негативний – також визначається як помилка II роду.

Дані значення і представлені у матриці невідповідності (рис. 4.15), а саме: ІП = 56219, ІН = 253, ХН = 438, ХП = 52. Маючи дані значення, можна оцінити модель більш повно. Також визначимо наступні відомі характеристики:

– позитивний стан П (condition positive) P – дійсна кількість позитивних випадків у даних;

– негативний стан Н (condition negative) N – дійсна кількість негативних випадків у даних.

У нашому випадку, як видно з матриці невідповідності, П = 56657, Н = 305.

Маючи вищевказані значення, визначимо:

– чутливість моделі (sensitivity), recall, true positive rate (TPR)

$$TPR = \frac{ІП}{П} = \frac{ІП}{ІП + ХН} = 1 - FRN = \frac{56219}{56657} = 0{,}9923 = 99{,}23\% \qquad (4.1)$$

– специфічність (specificity), selectivity, true negative rate (TNR)

$$TNR = \frac{ІН}{Н} = \frac{ІН}{ІН + ХП} = 1 - FPR = \frac{253}{305} = 0{,}8295 = 82{,}95\% \qquad (4.2)$$

– позитивне прогностичне значення – точність (precision), positive predictive value (PPV)

$$PPV = \frac{ІП}{ІП + ХП} = \frac{56219}{56219 + 52} = 0{,}9991 = 99{,}91\% \qquad (4.3)$$

– негативне прогностичне значення – negative predictive value (NPV)

$$NPV = \frac{ІН}{ІН + ХН} = \frac{253}{253 + 438} = 0{,}3661 = 36{,}61\% \quad (4.4)$$

– помилка другого роду – miss rate or false negative rate (FNR)

$$FNR = \frac{ХН}{П} = \frac{ХН}{ХН + ІП} = 1 - TRN = \frac{438}{56657} = 0{,}007791 = 0{,}78\% \quad (4.5)$$

– помилка першого роду – fall-out or false positive rate (FPR)

$$FPR = \frac{ХП}{Н} = \frac{ХП}{ХП + ІН} = 1 - TNR = \frac{52}{305} = 0{,}1705 = 17{,}05\% \quad (4.6)$$

– частка помилкових відхилень – false discovery rate (FDR)

$$FDR = \frac{ХП}{ХП + ІП} = 1 - PPV = \frac{52}{52 + 56219} = 0{,}0009 = 0{,}09\%; \quad (4.7)$$

– частка помилкових упущень – false omission rate (FOR)

$$FOR = \frac{ХН}{ХН + ІН} = 1 - NPV = \frac{438}{438 + 253} = 0{,}633863 = 63{,}39\% \quad (4.8)$$

– загальна точність – accuracy (ACC)

$$ACC = \frac{ІП + ІН}{П + Н} = \frac{ІП + ІН}{ІП + ІН + ХП + ХН} = \frac{56219 + 253}{56657 + 305} = 0{,}99139 = 99{,}14\% \quad (4.9)$$

– F1 score – F-міра – гармонійне середнє значення точності або чутливість

$$F1 = 2 * \frac{PPV * TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN} = 2 * \frac{0{,}9991 * 0{,}9923}{0.9991 + 0{,}9923} = 0{,}99569 = 99{,}57\% \quad (4.10)$$

– Matthews correlation coefficient (MCC) – Коефіцієнт кореляції Метьюса

$$MCC = \frac{ИП * ІН - ХН * ХП}{\sqrt{(ИП + ХН) * (ИП + ХП) * (ІН + ХН) * (ІН + ХП)}} =$$

$$= \frac{56219 * 253 - 438 * 52}{\sqrt{(56219 + 438) * (56219 + 52) * (253 + 438) * (253 + 52)}} = \frac{14200631}{\sqrt{56657 * 56271 * 691 * 305}} =$$

$$= 0{,}54783 = 54{,}78\% \quad (4.11)$$

– J статистика Йоудена Informedness or Bookmaker Informedness (BM)

$$BM = TPR + TNR - 1 = 0{,}9923 + 0{,}8295 - 1 = 0{,}8218 = 82{,}18\%; \quad (4.12)$$

– позначеність – markedness (MK)

$$MK = PPV + NPV - 1 = 0{,}9991 + 0{,}3661 - 1 = 0{,}3652 = 36{,}52\% \quad (4.13)$$

Зведемо отримані дані до рис. 4.17.

Здійснимо таку ж оцінку отриманої моделі та побудуємо матриці невідповідності без нормалізації (рис. 4.18) та з нормалізацією (рис. 4.19) на другій контрольній вибірці з користувачами-шахраями.

Оцінивши метрику похибок моделі, необхідно перевірити її адекватність.

Лістинг моделювання запропонованого в роботі методу виявлення шахрайства при інсталюванні мобільних додатків представлено у Додатку Б.

Для вирішення поставленої задачі виявлення шахрайства при інсталюванні мобільних додатків розглянемо, реалізуємо та порівняємо відомі методи класифікації.

		Експертна оцінка		
		Позитивний стан	Негативний стан	
Оцінка інформаційної технології	Позитивний результат	**Істинно позитивна** (ІП) = 56219	**Хибно позитивна** (ХП) = 52	Positive predictive value = ІП / (ІП + ХП) = 56219 / (56219 + 52) = **99,91%**
	Негативний результат	**Хибно негативна** (ХН) = 438	**Істинно негативна** (ІН) = 253	Negative predictive value = ІН / (ХН + ІН) = 253 / (438 + 253) **≈ 36,61%**
		Чутливість моделі = ІП / (ІП + ХН) = 56219 / (56219 + 438) **≈ 99,23%**	Specificity = ІН / (ХП + ІН) = 253 / (52 + 253) **= 82,95%**	

Рисунок 4.17 – Матриця невідповідності з обчисленими оцінками моделювання

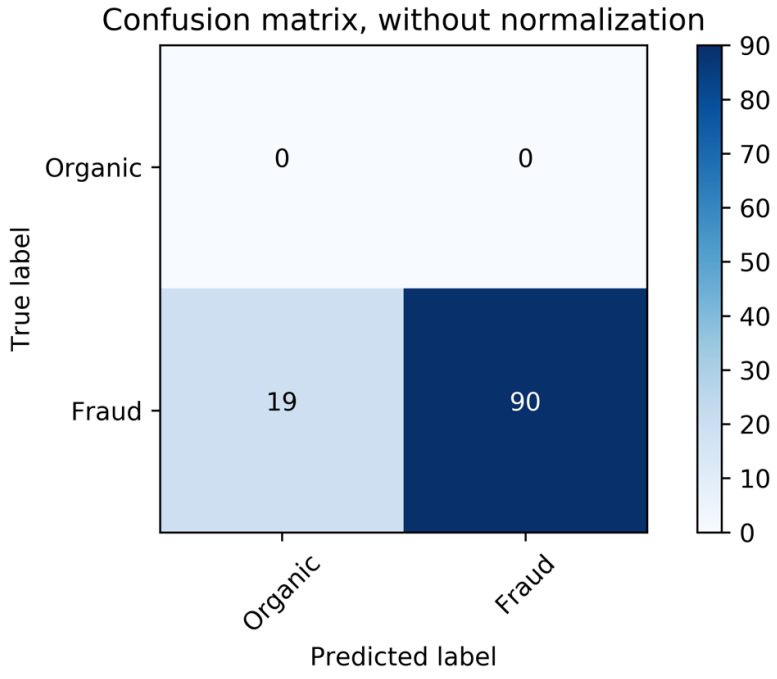


Рисунок 4.18 – Матриця невідповідності другої контрольної вибірки з шахраями без нормалізації

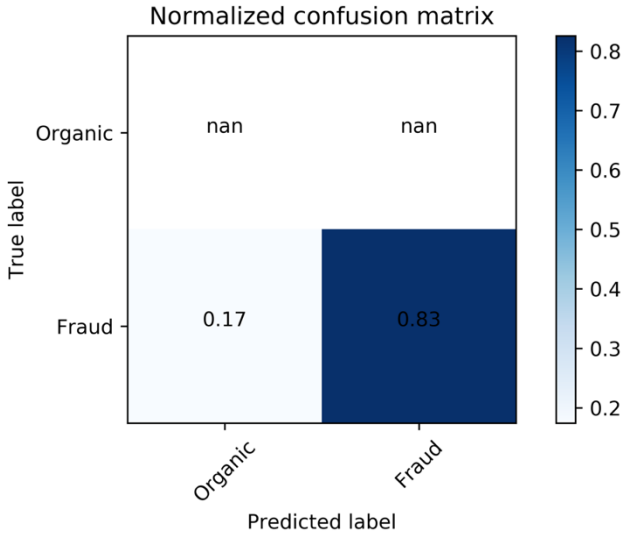


Рисунок 4.19 – Нормалізована матриця невідповідності другої контрольної вибірки з шахраями

При дослідженні способів реалізації шахрайських операцій під час інсталювання мобільних додатків, було виділено наступні варіанти:

- кліковий спам (Click Spamming),
- мобільне викрадення (Mobile Hijacking),
- ферми дій (Action Farms),
- запуск скриптів для будь-якої кількості інсталяцій додатку.

Для перевірки методів класифікації, нам необхідно виділити класи. Назвемо їх відповідно до способу інсталяції:

- «Organic» (органічні інсталяції),
- «Fraudulent» (шахрайські інсталяції).

Отже, реалізуємо декілька алгоритмів машинного навчання, використовуючи Python та Scikit-learn – найпопулярніший інструмент для машинного навчання на Python. Використаємо репрезентативний набір даних для навчання класифікатора, задачею якого буде навчитися правильно визначати спосіб інсталяції – органічний чи шахрайський. **Метою даного експерименту є** порівняння результатів моделювання існуючих підходів для виявлення шахрайства при інсталюванні мобільних додатків та запропонованого у роботі

методу виявлення шахрайства при інсталюванні мобільних додатків, і визначення найбільш ефективного з них.

Отже, на основі зібраних даних створимо навчальні та тестові набори даних, застосуємо масштабування та побудуємо наступні моделі:

– логістична регресія, програмна реалізація з використанням мови програмування Python та результати якої представлено на рис. 4.20. Моделювання показало, що дана модель на тренувальній вибірці дала точність 0,70, а на тестовій – 0,40;

Logistic Regression

```
In [4]: from sklearn.linear_model import LogisticRegression

        logreg = LogisticRegression()
        logreg.fit(X_train, y_train)

        print('Accuracy of Logistic regression classifier on training set: {:.2f}'
             .format(logreg.score(X_train, y_train)))
        print('Accuracy of Logistic regression classifier on test set: {:.2f}'
             .format(logreg.score(X_test, y_test)))
```

```
Accuracy of Logistic regression classifier on training set: 0.70
Accuracy of Logistic regression classifier on test set: 0.40
```

Рисунок 4.20 – Результати моделювання моделі логістичної регресії з використанням мови програмування Python та точність результатів моделювання

– дерево рішень, програмна реалізація з використанням мови програмування Python та результати якої представлено на рис. 4.21. Моделювання показало, що дана модель на тренувальній вибірці дала точність 0,98, а на тестовій – 0,72;

Decision Tree

```
In [10]: from sklearn.tree import DecisionTreeClassifier

         clf = DecisionTreeClassifier().fit(X_train, y_train)

         print('Accuracy of Decision Tree classifier on training set: {:.2f}'
              .format(clf.score(X_train, y_train)))
         print('Accuracy of Decision Tree classifier on test set: {:.2f}'
              .format(clf.score(X_test, y_test)))
```

```
Accuracy of Decision Tree classifier on training set: 0.98
Accuracy of Decision Tree classifier on test set: 0.72
```

Рисунок 4.21 – Програмна реалізація моделі дерева рішень з використанням мови програмування Python та точність результатів моделювання

– k-найближчих сусідів, програмна реалізація з використанням мови програмування Python та результати якої представлено на рис. 4.22. Моделювання показало, що дана модель на тренувальній вибірці дала точність 0,94, а на тестовій – 0,98;

K-Nearest Neighbors

```
In [10]: from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
print('Accuracy of K-NN classifier on training set: {:.2f}'
     .format(knn.score(X_train, y_train)))
print('Accuracy of K-NN classifier on test set: {:.2f}'
     .format(knn.score(X_test, y_test)))
```

```
Accuracy of K-NN classifier on training set: 0.94
Accuracy of K-NN classifier on test set: 0.98
```

Рисунок 4.22 – Програмна реалізація методу k-найближчих сусідів з використанням мови програмування Python та точність результатів моделювання

– лінійний дискримінантний аналіз, програмна реалізація з використанням мови програмування Python та результати якої представлено на рис. 4.23. Моделювання показало, що дана модель на тренувальній вибірці дала точність 0,80, а на тестовій – 0,66;

Linear Discriminant Analysis

```
In [14]: from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)
print('Accuracy of LDA classifier on training set: {:.2f}'
     .format(lda.score(X_train, y_train)))
print('Accuracy of LDA classifier on test set: {:.2f}'
     .format(lda.score(X_test, y_test)))
```

```
Accuracy of LDA classifier on training set: 0.80
Accuracy of LDA classifier on test set: 0.66
```

Рисунок 4.23 – Програмна реалізація лінійного дискримінантного аналізу з використанням мови програмування Python та точність результатів моделювання

– Гаусовий Байєс, програмна реалізація з використанням мови програмування Python та результати якої представлено на рис. 4.24. Моделювання показало, що дана модель на тренувальній вибірці дала точність 0,80, а на тестовій – 0,66;

Gaussian Naive Bayes

```
In [14]: from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()
gnb.fit(X_train, y_train)
print('Accuracy of GNB classifier on training set: {:.2f}'
     .format(gnb.score(X_train, y_train)))
print('Accuracy of GNB classifier on test set: {:.2f}'
     .format(gnb.score(X_test, y_test)))
```

```
Accuracy of GNB classifier on training set: 0.80
Accuracy of GNB classifier on test set: 0.66
```

Рисунок 4.24 – Програмна реалізація Гаусового Байєса з використанням мови програмування Python та точність результатів моделювання

– метод опорних векторів, програмна реалізація з використанням мови програмування Python та результати якої представлено на рис. 4.25. Моделювання показало, що дана модель на тренувальній вибірці дала точність 0,62, а на тестовій – 0,34.

Support Vector Machine

```
In [15]: from sklearn.svm import SVC

svm = SVC()
svm.fit(X_train, y_train)
print('Accuracy of SVM classifier on training set: {:.2f}'
     .format(svm.score(X_train, y_train)))
print('Accuracy of SVM classifier on test set: {:.2f}'
     .format(svm.score(X_test, y_test)))
```

```
Accuracy of SVM classifier on training set: 0.62
Accuracy of SVM classifier on test set: 0.34
```

Рисунок 4.25 – Програмна реалізація методу опорних векторів з використанням мови програмування Python та точність результатів моделювання

Результати моделювання запропонованого методу виявлення шахрайства при інсталюванні мобільних додатків представлено на рис. 4.19.

Здійснивши порівняльний аналіз точності та швидкодії розробленої інформаційної технології та систем-аналогів виявлення шахрайства, зведемо отримані результати до таблиць 4.1 та 4.2 відповідно. Для порівняння було взято системи-аналоги, які найчастіше використовуються в даній предметній області. Порівняльний аналіз проводився на вибірці, наданій експертами.

Таблиця 4.1

Порівняльний аналіз точності розробленої інформаційної технології та систем-аналогів виявлення шахрайства

Метод	**Точність виявлення класу користувача на тестовій вибірці, %**	**Точність виявлення класу шахрая на тестовій вибірці, %**
Запропонований метод виявлення шахрайства при інсталюванні мобільних додатків	99,14	82,95
Fraudlogix	97,88	81,4
Kochava	90,02	79,48
Appsflyer	95,97	80,14

Результати порівняння розробленого узагальненого методу виявлення шахрайства, який покладено в основу інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків, та існуючих методів для виявлення шахрайства, зведено до таблиці 4.3.

В таблиці 4.2 показано, на скільки швидкість виявлення шахрайства при інсталюванні мобільних додатків з використанням запропонованого методу вища за швидкість виявлення шахрайства при інсталюванні мобільних додатків з використанням систем-аналогів.

Таблиця 4.2

Порівняльний аналіз швидкодії розробленої інформаційної технології та систем-аналогів виявлення шахрайства

Метод	**Підвищення середньої швидкості виявлення шахрайства на тестовій вибірці запропонованого методу виявлення шахрайства на, %**
Fraudlogix	1,78 %
Kochava	2,0082 %
Appsflyer	5,83 %

Таблиця 4.3

Порівняльний аналіз ефективності розробленого узагальненого методу виявлення шахрайства та існуючих методів, що використовуються для виявлення шахрайства

Метод	**Точність моделі класифікації на тренувальній вибірці**	**Точність моделі класифікації на тестовій вибірці**
Запропонований метод виявлення шахрайства при інсталюванні мобільних додатків	0,9962	0,9914
Logistic Regression	0,70	0,40
Decision Tree	0,98	0,72
K-Nearest Neighbors	0,94	0,98
Linear Discriminant Analysis	0,80	0,66
Gaussian Naive Bayes	0,80	0,66
Support Vector Machine	0,62	0,34

4.5 Аналіз результатів впровадження

Ефективне застосування та впровадження розробленої інформаційної технології передбачає попередній аналіз предметної області виявлення шахрайства при інсталюванні мобільних додатків та її розробку. Підсистема тренування класифікаційної моделі та підсистема класифікації не потребують попередньої трудомісткої роботи з аналізу предметної області, але передбачають наявність вектору однорідних даних, отриманого з підсистеми подолання різнорідності даних. Підсистема виявлення характеристик користувача, у свою чергу, передбачає попередній аналіз предметної області та наявність початкового поміченого набору даних про користувачів при інсталюванні мобільних додатків.

4.5.1 Впровадження інформаційної технології для виявлення шахрайства при інсталюванні мобільних додатків в ТОВ «ВІН ІНТЕРАКТИВ»

Результати дисертаційних досліджень були впроваджені та отримали практичну реалізацію в ТОВ «ВІН ІНТЕРАКТИВ» (Додаток А) [13]-[16], [18]-[28]. А саме: алгоритми процесу подолання різнорідності даних та модель процесу подолання різнорідності даних, що в сукупності дозволило всю множину різнорідних даних про користувачів звести до вектору однорідних даних. Використання отриманого вектору однорідних даних дозволило здійснити класифікацію користувачів на класи «шахрай» та «органічний користувач» з точністю на 1,26 % вищу, ніж з використанням існуючих систем-аналогів, зокрема Fraudlogix, яка показувала найкращі результати при виявленні шахрайства при інсталюванні мобільних додатків у даній компанії.

4.5.2 Впровадження інформаційної технології в Garuda AI B.V.

Результати дисертаційних досліджень були впроваджені та отримали практичну реалізацію в компанії “Garuda AI B.V.”, яка займається розробкою різноманітного програмного забезпечення, пов’язаного з виявленням шахрайських атак у мобільних додатках, які в основному працюють з блокчейном (Додаток А).

До інтерфейсу керування користувачами та їх маркетинговими кампаніями, а також їх класами (шахрайські чи органічні) висувають високі вимоги. Він повинен бути зручним, інтуїтивно зрозумілим, а також має містити потужні компоненти керування структурою даних. Остання вимога дуже важлива, оскільки від якості структури вебсторінки залежить на скільки зручно та швидко адміністратори вебсайту дізнаються про появу шахрайських користувачів чи маркетингових кампаній та зрозуміють, яким маркетинговим кампаніям не потрібно платити кошти за інсталяції та вказати їм при цьому точну причину, із-за якої користувачі та маркетингові кампанії були помічені як шахрайські. Виходячи з цього, керівництво компанії Garuda AI B.V. прийняло рішення підвищити ефективність системи управління користувачами, маркетинговими кампаніями та їх класами (шахрайський або органічний) шляхом інтеграції в неї розробленої технології.

Результати дисертаційних досліджень впроваджені у вигляді створеного в роботі програмного забезпечення, що містить в собі підсистему виявлення характеристик користувача; підсистему подолання різнорідності даних; підсистему тренування класифікаційної моделі; підсистему класифікації; підсистему формування бази даних шахраїв; підсистему формування бази знань (для виявлення шахраїв); підсистему інтелектуального аналізу даних та формування портрету користувачів; підсистему прогнозування узагальненого шаблону шахрая та інтерфейс перегляду й налаштування ІТ.

За рахунок впровадження ІТ було досягнуто позитивного ефекту – система управління контентом Garuda AI B.V. з розробленими підсистемами виявлення

характеристик користувача, подолання різнорідності даних, тренування класифікаційної моделі, класифікації, формування бази даних шахраїв, формування бази знань (для виявлення шахраїв), інтелектуального аналізу даних та формування портрету користувачів, прогнозування узагальненого шаблону шахрая успішно функціонує на ряді корпоративних ресурсів в режимі реального часу, що дозволяє адміністраторам більш точно визначати класи користувачів та маркетингових кампаній, дізнаватись причину їх зарахування до класу шахраїв та зменшити завдяки цьому економічні витрати.

Впровадження запропонованої в роботі інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків, на відміну від існуючих систем Antifraud, дозволило підвищити точність виявлення класу користувача до 99,14 %, точність виявлення класу шахрая – до 82,95 %.

4.5.3 Впровадження інформаційної технології у ТОВ «4ХайТек»

Результати дисертаційних досліджень були впроваджені та отримали практичну реалізацію в компанії ТОВ «4ХайТек» [16], а саме для виявлення шахрайства та причини його виникнення при інсталюванні мобільних додатків на основі бази даних, що складається із подій 205689 користувачів (Додаток А).

Результати дисертаційних досліджень впроваджені у вигляді створеного в роботі програмного забезпечення, яке включає в себе модель процесу подолання різнорідності вхідних даних, алгоритм виявлення шахрая при інсталюванні мобільних додатків, алгоритм створення узагальненого портрету шахрая. В роботі розроблено додаток у вигляді консольних утиліт, що надають інформацію щодо класу користувачів та маркетингових кампаній і причину зарахування їх до класу «шахрай».

За рахунок впровадження ІТ було досягнуто позитивного ефекту, точність визначення шахраїв підвищилась на 2,81 %, завдяки чому витрати за інсталяції від маркетингових кампаній скоротились на 25,67%, а швидкість пошуку інформації підвищилася на 5,83% завдяки створенню узагальненого портрету шахрая.

4.5.4 Впровадження інформаційної технології у ПП «Літсофт»

Результати дисертаційних досліджень були впроваджені та отримали практичну реалізацію в компанії ПП «Літсофт» [16], а саме для створення вебсайту, у якому можна визначити класи («шахрайський», «органічний») користувачів і маркетингових кампаній та причини шахрайства при інсталюванні мобільних додатків, а також мобільного додатку із функцією сповіщення про появу шахрайського користувача або маркетингової кампанії на основі бази даних, що складається із подій 623009 користувачів (Додаток А).

Результати дисертаційних досліджень впроваджені у вигляді створеного в роботі програмного забезпечення, яке включає в себе алгоритм подолання різнорідності даних, узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків, алгоритм створення узагальненого портрету шахрая. В роботі розроблено вебсайт та мобільний додаток, вимоги до яких представлено у підрозділі 3.6.

За рахунок впровадження ІТ було досягнуто позитивного ефекту, точність визначення шахраїв підвищилась на 3,47 %, завдяки чому витрати за інсталяції від маркетингових кампаній скоротились на 19,4%, а швидкість пошуку інформації підвищилася на 2,0082% завдяки створенню узагальненого портрету шахрая.

4.5.5 Впровадження інформаційної технології у навчальний процес

У 2016-19 роках на кафедрі комп'ютерних наук Вінницького національного технічного університету впроваджено у навчальний процес результати дисертаційної роботи на здобуття наукового ступеня кандидата технічних наук, отримані Польгуль Т.Д., а саме:

– інформаційна технологія виявлення шахрайства при інсталюванні мобільних додатків;

– узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків;

– моделі та алгоритми процесу подолання різнорідності вхідних даних.

Теоретичні результати дисертаційної роботи використано в навчальному процесі на кафедрі комп'ютерних наук ВНТУ при підготовці фахівців зі спеціальності 8.05010104 – Системи штучного інтелекту (напрям 6.050101 – Комп’ютерні науки), 122 – Комп’ютерні науки при викладанні таких дисциплін, як «Інтелектуальний аналіз даних», «Методи та системи штучного інтелекту», «Основи проектування систем штучного інтелекту», «Проектування інформаційних систем», «Технології розподілених систем та паралельних обчислень», «Сучасні інформаційні технології в науці та освіті».

4.6 Висновки до розділу 4

Розроблено методику проведення експериментальних досліджень розробленої інформаційної технології.

Розроблено алгоритм мінімізації часу виявлення шахраїв на основі розпаралелення обчислювальних процесів.

Доведено адекватність моделей процесу подолання різнорідності вхідних даних та моделі класифікації вхідних даних на вибірці з розробленого мобільного додатку «MobNsters: Mafia War Strategy» (Orneon Ltd.) [16], [17]-[19], [28], [92]-[94], яку було поділено на три набори даних – тренувальний, перевірочний (валідаційний) та тестовий (контрольний). Сформовано графіки втрат (loss) та точності (accuracy) на етапах навчання та перевірки моделей (рис. 8). Для уникнення перенавчання здійснено навчання моделі у 100 епох. Побудовано нормалізовану та ненормалізовану матрицю невідповідності (рис. 9), на основі якої визначено чотири результати – істинно позитивний, істинно негативний, хибно позитивний, хибно негативний, а також: чутливість моделі – 99,23 %, специфічність – 82,95 %, позитивне прогностичне значення – 99,91 %, негативне прогностичне значення – 36,61 %, помилка першого роду – 17,05 %, помилка другого роду – 0,78%, частка помилкових відхилень – 0,09 %, частка помилкових упущень – 63,39%, загальна точність –

99,14 %, F-міра – 99,57 %, коефіцієнт кореляції Метьюса – 54,78 %, J статистика Йоудена – 82,18 %, позначеність – 36,52 %. Визначено, що точність методу виявлення шахрайства, який працює з даними, отриманими методом подолання різнорідності вхідних даних, рівна 99,14%. Наявних шахраїв метод визначає з точністю 82,76%, загалом точність визначення класу користувачів – 99,14%. Результати роботи програмного забезпечення подано в дисертаційній роботі. Визначено критерії ефективності та здійснено аналіз результатів тестування. Здійснено порівняльний аналіз ефективності розробленого узагальненого методу виявлення шахрайства та існуючих методів, що використовуються для виявлення шахрайства. Точність запропонованої удосконаленої моделі класифікації на тестовій вибірці на 1,14 % вища від найкращої із змодельованих моделей класифікації. Здійснено порівняльний аналіз точності та швидкодії розробленої інформаційної технології та систем-аналогів виявлення шахрайства. Показано, що точність виявлення класу користувача з використанням розробленої технології було підвищено на 1.26%, а точність виявлення шахрая підвищено на 1.36% у порівнянні з системою-аналогом, що дала найкращий результат.

На основі запропонованої в роботі інформаційної технології розроблено модульне програмне забезпечення “Mobile App Install Fraud Detection System” на основі алгоритмів, які реалізують розроблені методи, моделі та веб-систему.

Розроблене модульне програмне забезпечення було впроваджено на підприємствах:

– Garuda AI B.V. (Нідерланди) – інформаційна технологія;

– ТОВ «ВІН ІНТЕРАКТИВ» – алгоритми подолання різнорідності даних, модель процесу подолання різнорідності даних;

– ТОВ «4ХайТек» – модель процесу подолання різнорідності вхідних даних, методика виявлення шахрая при інсталюванні мобільних додатків;

– ПП «Літсофт» – алгоритм подолання різнорідності даних, узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків, методика створення узагальненого портрету шахрая;

– у навчальний процес кафедри КН Вінницького національного технічного університету – узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків;

– у навчальний процес кафедри інформатики, програмної інженерії та економічної кібернетики Херсонського державного університету – інформаційна технологія виявлення шахрайства при інсталювання мобільних додатків з використанням інтелектуального аналізу даних.

Результати проведених досліджень опубліковані в [15], [17], [18], [23], [24], [28], [63].

ВИСНОВКИ

У дисертаційній роботі на основі виконаних теоретичних і експериментальних досліджень розв’язано актуальне наукове завдання підвищення точності та швидкодії процесу виявлення шахраїв шляхом розробки узагальненого методу виявлення шахрайства при інсталюванні мобільних додатків, методу подолання різнорідності вхідних даних, удосконаленої моделі класифікації користувачів на основі глибинних нейронних мереж, алгоритмів процесу подолання різнорідності вхідних даних, алгоритму створення узагальненого портрету шахрая та алгоритму пошуку аномалій в даних, які покладені в основу запропонованої інформаційної технології.

При цьому отримано такі основні результати:

1. Здійснений аналіз об’єкту дослідження дозволив визначити задачу виявлення шахрайства як одну із задач пошуку аномалій в даних.

2. Здійснений аналіз методів виявлення шахрайства при інсталюванні мобільних додатків та їх класифікація дозволили визначити методи машинного навчання для вирішення поставлених задач.

3. Формалізація процесу виявлення шахрайства як аномалії в даних з використанням теорії множин дозволила визначити властивості аномальних і неаномальних даних у визначеній предметній області та спростити процес пошуку шахрайства при інсталюванні мобільних додатків.

4. Здійснена класифікація різнорідних даних при інсталюванні мобільних додатків на основі процесу вибору ознак дозволила спростити процес аналізу різнорідних за метриками, розмірностями і шаблонами даних та автоматизувати його.

5. Вперше розроблено узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків, відмінність якого полягає у використанні запропонованої моделі класифікації користувачів та методу подолання різнорідності вхідних даних, що дозволяє визначити класи користувачів та підвищити точність виявлення шахрайства при інсталюванні мобільних додатків.

6. Вперше запропоновано метод подолання різнорідності вхідних даних, що являє собою сукупність процедур вибору ознак, зниження розмірності та нормалізації даних, відмінність якого полягає у новій моделі процесу подолання різнорідності даних шляхом шкалювання за інформативністю, що дозволяє всю множину різнорідних даних про користувачів звести до вектору уніфікованих ознак без зменшення діагностичної цінності інформації.

7. Удосконалено модель класифікації користувачів на основі глибинних нейронних мереж у частині зниження розмірності та нормалізації даних згідно запропонованого методу подолання різнорідності даних, яка є основою для створення узагальненого портрету шахрая з метою спрощення процесів їх виявлення.

8. Розроблено моделі інформаційної технології з використанням технологій системного аналізу та моделювання, а саме: побудована логічна, концептуальна моделі інформаційної технології. Здійснено аналіз та розробку структур даних інформаційної технології, розроблено шаблон шахрая для формування портрету шахрая. Як показали дослідження, усі розроблені моделі є необхідними для ефективного функціонування інформаційної технології виявлення шахрайства на основі запропонованого в роботі узагальненого методу, в основі якого лежать моделі, методи та алгоритми виявлення аномалій в даних.

9. Розроблено алгоритми процесу подолання різнорідності вхідних даних, алгоритм створення узагальненого портрету шахрая та алгоритм пошуку аномалій в даних, які покладені в основу запропонованої інформаційної технології, що підвищило точність та швидкодію процесу виявлення шахраїв.

10. Розроблено алгоритм пошуку аномалій в даних як основа функціонування інформаційної технології за допомогою діаграми activity, а також здійснено аналіз процесів в інформаційній технології виявлення шахрайства за допомогою UML-діаграми послідовності.

11. Запропоновано інформаційну технологію виявлення шахрайства при інсталюванні мобільних додатків, яка використовує запропоновані метод виявлення шахрайства при інсталюванні мобільних додатків, метод подолання

різнорідності вхідних даних, модель класифікації даних, і, на відміну від існуючих систем Antifraud, дозволила підвищити точність класифікації користувачів до 99,14 %, зокрема точність класифікації шахраїв – до 82,76 %.

12. Розроблено методику проведення експериментальних досліджень розробленої інформаційної технології.

13. Розроблено алгоритм мінімізації часу виявлення шахраїв на основі розпаралелення обчислювальних процесів.

14. Доведено адекватність моделей процесу подолання різнорідності вхідних даних та моделі класифікації вхідних даних на вибірці з розробленого мобільного додатку «MobNsters: Mafia War Strategy» (Orneon Ltd.). Зокрема показано, що точність виявлення класу користувача з використанням розробленої технології було підвищено на 1.26%, а точність виявлення шахрая підвищено на 1.36% у порівнянні з системою-аналогом, що дала найкращий результат.

15. На основі запропонованої в роботі інформаційної технології розроблено модульне програмне забезпечення “Mobile App Install Fraud Detection System” з використанням алгоритмів, які реалізують розроблені методи, моделі та веб-систему, що впроваджена на підприємстві Garuda AI B.V. (Нідерланди).

16. Результати дисертаційної роботи та розроблене модульне програмне забезпечення впроваджені на підприємствах Garuda AI B.V. (Нідерланди) – інформаційна технологія; ТОВ «ВІН ІНТЕРАКТИВ» – алгоритми подолання різнорідності даних, модель процесу подолання різнорідності даних; ТОВ «4ХайТек» – модель процесу подолання різнорідності вхідних даних, методика виявлення шахрая при інсталюванні мобільних додатків; ПП «Літсофт» – алгоритм подолання різнорідності даних, узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків, методика створення узагальненого портрету шахрая; у навчальний процес кафедри КН Вінницького національного технічного університету – узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків та у навчальний процес кафедри інформатики, програмної інженерії та економічної кібернетики

Херсонського державного університету – інформаційна технологія виявлення шахрайства при інсталювання мобільних додатків з використанням інтелектуального аналізу даних. Впровадження результатів дисертаційних досліджень підтверджено відповідними актами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] “Our take on mobile fraud detection”, available at: http://geeks.jampp.com/data-science/mobile-fraud/

[2] Vacha Dave, Saikat Guha, and Yin Zhang, “ViceROI: Catching Click-Spam in Search Ad Networks”, available at: http://www.sysnet.ucsd.edu/~vacha/ccs13.pdf

[3] V. Dave, S. Guha, and Y. Zhang, “Measuring and Fingerprinting Click-Spam in Ad Networks”, in *Proc. Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)”,* Helsinki, Finland, 2012, pp. 175-186.

[4] Fraudlogix: Ad Fraud Solutions for Exchanges, Networks, SSPs & DSPs. [Online]. Available: https://www.fraudlogix.com/

[5] Kraken Antibot. [Online]. Available: http://kraken.run/

[6] Adjust . [Online]. Available: https://www.adjust.com/

[7] Kochava Uncovers Global Ad Fraud Scam . [Online]. Available: https://www.kochava.com/

[8] TMC Attribution Analytics. [Online]. Available: https://help.tune.com/marketing-console/attribution-analytics/

[9] Appsflyer: Protect your data from mobile fraud: Protect360. [Online]. Available: https://www.appsflyer.com/product/protect360/

[10] AppsFlyer: Measure In-App To Grow Your Mobile Business. [Online]. Available: https://www.appsflyer.com/

[11] FraudScore: FraudScore fights ad fraud using Machine Learning. [Online]. Available: https://fraudscore.mobi/

[12] AppMetrica. [Online]. Available: https://appmetrica.yandex.ru/

[13] Т. Д. Польгуль, та А. А. Яровий, “Метод подолання різнорідності даних для виявлення шахрайства при інсталюванні мобільних додатків”, *Вісник СНУ ім. В. Даля – Сєвєродонецьк: СНУ ім. В. Даля,* № 7 (248), с.60-69, 2018.

[14] T. Polhul, and A. Yarovyi, “Development of a method for fraud detection in heterogeneous data during installation of mobile applications”, *Eastern-European Journal of Enterprise Technologies*, № 1/2 (97), 2019. doi: 10.15587/1729-

4061.2019.155060

[15] Т.Д. Польгуль, та А.А. Яровий, “Аналіз різнорідних даних в інтелектуальних системах виявлення шахрайства”, *Вісник Вінницького політехнічного інституту*, № 2, с. 78-90, 2019.

[16] Т.Д. Польгуль, “Інформаційна технологія побудови інтелектуальних систем виявлення шахрайства при інсталюванні мобільних додатків”, *Інформаційні технології та комп'ютерна інженерія*, № 1, с. 4-16, 2019.

[17] A. Yarovyi, and T. Polhul, “Applied Aspects of Implementation of Intelligent Information Technology for Fraud Detection During Mobile Applications Installation”, *Advances in Intelligent Systems and Computing IV. CCSIT 2019: Advances in Intelligent Systems and Computing*, Springer, Cham, Switzerland, vol 1080, pp. 377-386, 2019. doi: https://doi.org/10.1007/978-3-030-33695-0_26

[18] T. Polhul, “Conceptual Model of an Intelligent System for Detecting Fraud During Mobile Applications Installation”, in *Proc. 10th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Leeds, United Kingdom, pp. 167-174, 2019. doi: 10.1109/DESSERT.2019.8770030. Режим доступу: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8770030&isnumber=8770005.

[19] А.А. Яровий, О.Н. Романюк, І.Р. Арсенюк, та Т.Д. Польгуль, “Виявлення шахрайства при інсталюванні програмних додатків з використанням інтелектуального аналізу даних”, *Наукові праці Донецького національного технічного університету. Серія “Інформатика, кібернетика та обчислювальна техніка”*, № 2 (25), с. 126-131, 2017. Режим доступу: http://science.donntu.edu.ua/wp-content/uploads/2018/03/ikvt_2017_2_site-1.pdf

[20] Т.Д. Польгуль, та А.А. Яровий, “Визначення шахрайських операцій при встановленні мобільних додатків з використанням інтелектуального аналізу даних”, *Сучасні тенденції розвитку системного програмування. Тези доповідей*, Київ, 2016, с. 55-56. Режим доступу: http://ccs.nau.edu.ua/wp-content/uploads/2017/12/%D0%A1%D0%A2%D0%A0%D0%A1%D0%9F_2016_07.pdf

[21] Т.Д. Польгуль, та А.А. Яровий, “Визначення шахрайських операцій при інсталяції мобільних додатків з використанням інтелектуального аналізу даних”, на *XLVI науково-технічній конференції підрозділів ВНТУ,* Вінниця, 2017. Режим доступу: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/17200/2158.pdf?sequence=3

[22] А. Яровий, Т. Польгуль, та Л. Крилик, “Розробка методу виявлення шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних”, *XIV Міжнародна конференція Контроль і управління в складних системах (КУСС-2018). Тези доповідей*, Вінниця, 2018, с. 35.

[23] T. Polhul, “Development of an intelligent system for detecting mobile app install fraud”, in *Proc.IRES 156th International Conference*, Bangkok, Thailand, 2019, pp. 25-29.

[24] T. Polhul, “Development of an intelligent system for detecting mobile app install fraud”, *International Journal of Advances in Electronics and Computer Science (IJAECS)*, vol. 6, no. 7, pp. 13–17, July, 2019.

[25] А.А. Яровий, та Т.Д. Польгуль, “Подолання різнорідності вхідних даних при виявленні шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних”, на *П’ятій міжнародній науково-технічній конференції студентів, магістрів, аспірантів «Інформатика, управління та штучний інтелект»,* Національний технічний університет «Харківський політехнічний інститут», Харків, 2018, с. 109.

[26] T. Polhul and A. Yarovyi, "Method of Fraudster Fingerprint Formation During Mobile Application Installations", *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Metz, France, 2019, pp. 1099-1103. doi: 10.1109/IDAACS.2019.8924369. Режим доступу: https://ieeexplore.ieee.org/document/8924369.

[27] Т.Д. Польгуль, “Моделювання процесу виявлення шахрайства при інсталюванні мобільних додатків”, на *XLVIII науково-технічної конференції*

підрозділів ВНТУ, Вінниця, 2019. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2019/paper/view/6863

[28] A. Yarovyi, and T. Polhul, “Intelligent information technology for fraud detection during mobile applications installation”, in *Proc. 14th International conference "Computer sciences and Information technologies" (CSIT 2019)*, pp. 1-5, Lviv, 2019. doi: 10.1109/STC-CSIT.2019.8929827. Режим доступу: https://ieeexplore.ieee.org/ document/8929827.

[29] Impact: Forensiq by Impact Earns MRC Accreditation for SIVT Detection and Filtration and Viewability Measurement. [Online]. Available: https://impact.com/ad-fraud-detection/

[30] V. Chandola, A. Banerjee, and V. Kumar, *Anomaly detection*. Minnesota, USA: ACM Computing Surveys, 2009, vol. 41, issue 3, pp. 1-58. doi: https://doi.org/10.1145/1541880.1541882

[31] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. USA: Aurélien Géron, O’Reilly Media, 2017.

[32] D. Cielen, A. D. B. Meysman, and M. Ali, *Introducing Data Science: Big data, machine learning, and more, using Python tools.* USA: Manning, 2016.

[33] S. Guido, and A. Müller, *Introduction to Machine Learning with Python: A Guide for Data Scientists*. USA: O’Reilly Media, 2016.

[34] F. Chollet, *Deep Learning with Python.* USA: Manning, 2017.

[35] D. Hawkins, *Identification of Outliers*. Chapman and Hall, 1980.

[36] Andrew Ng, *Machine Learning*. Standford, USA. [Online]. Available: https://www.coursera.org/learn/machine-learning?utm_source=gg&utm_medium=sem&utm_content=07-StanfordML-ROW&campaignid=2070742271&adgroupid=80109820241&device=c&keyword=machine%20learning%20mooc&matchtype=b&network=g&devicemodel=&adpostion=1t1&creativeid=369041663186&hide_mobile_promo&gclid=CjwKCAjwpuXpBRAAEiwAyRRPge3ilrZykf2wwoDbEgg4mLES-edSyJsV9n-r8QrtmNuE-WozUpviNBoCW6oQAvD_BwE

[37] X. Song, M. Wu, C. Jermaine, and S. Ranka, “Conditional Anomaly

Detection", *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, iss. 5, pp. 631-645, 2015. doi: https://doi.org/10.1109/tkde.2007.1009

[38] А. В. Гриценко, *Типы аномалий в видеоизображениях*. *Технические науки – от теории к практике: сборник статей по материалам VII международной научно-практической конференции. Часть I*. Новосибирск, Россия: СибАК, 2012. Режим доступа: https://sibac.info/conf/tech/vii/26730

[39] M. A. Prado-Romero, and A. Gago-Alonso, "Detecting contextual collective anomalies at a Glance", in *Proc. 2016 23rd International Conference on Pattern Recognition (ICPR)*, Cancun, Mexico, 2016. doi: https://doi.org/10.1109/ icpr.2016.7900017

[40] R. Agrawal, and R. Srikant, "Mining sequential patterns", in *Proc. Eleventh International Conference on Data Engineering*, Taipei, Taiwan, 1995. doi: https://doi.org/10.1109/icde.1995.380415

[41] D. Agarwal, "An Empirical Bayes Approach to Detect Anomalies in Dynamic Multidimensional Arrays", in *Proc. Fifth IEEE International Conference on Data Mining (ICDM'05)*, Houston, USA, 2005. doi: https://doi.org/10.1109/ icdm.2005.22

[42] C. Siaterlis, and B. Maglaris, "Towards multisensor data fusion for DoS detection", in *Proc. 2004 ACM symposium on Applied computing – SAC '04*, Nicosia, Cyprus, 2004. doi: https://doi.org/10.1145/967900.967992

[43] D. Agarwal, "Detecting anomalies in cross-classified streams: a Bayesian approach*", Knowledge and Information Systems*, vol. 11, iss. 1, pp. 29-44, 2006. doi: https://doi.org/10.1007/s10115-006-0036-4

[44] MachineLearning.ru. Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу даннях. [Электронный ресурс]. Режим доступу: http://www.machinelearning.ru

[45] T. Segaran, *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. Sebastopol, USA: O'Reilly Media, 2008.

[46] D.-Y. Yeung, and C. Chow, "Parzen-window network intrusion detectors*",* in *Proc. Object recognition supported by user interaction for service robots*, Hong Kong,

2002. doi: https://doi.org/10.1109/icpr.2002.1047476

[47] V. J. Hodge, and J. Austin, “A Survey of Outlier Detection Methodologies”, *Artificial Intelligence Review.*, vol. 22, issue 2, pp. 85-126. 2004. doi: https://doi.org/10.1007/s10462-004-4304-y

[48] Agyemang M., Barker K., and Alhajj R., “A comprehensive survey of numeric and symbolic outlier mining techniques*”, Intelligent Data Analysis*, vol. 10, iss. 6, pp. 521-538, 2006. doi: https://doi.org/10.3233/ida-2006-10604

[49] Keogh E., Lin J., and Fu A., “HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence”, *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Houston, USA, 2005. doi: https://doi.org/10.1109/icdm.2005.79

[50] E. Keogh, J. Lin, S.-H. Lee, and H. V. Herle, “Finding the most unusual time series subsequence: algorithms and applications”, *Knowledge and Information Systems*, vol. 11, iss. 1, pp. 1-27, 2006. doi: https://doi.org/10.1007/s10115-006-0034-6

[51] Donoho S., “Early detection of insider trading in option markets, in *Proc. Tenth ACM SIGKDD international conference on knowledge discovery and data mining – KDD-2004*, USA, 2004. doi: https://doi.org/10.1145/1014052.1014100

[52] A. W. Fu, O. T.-W. Leung, E. Keogh, and J. Lin, “Finding Time Series Discords Based on Haar Transform*”, Lecture Notes in Computer Science*, 2006, pp. 31–41. doi: https://doi.org/10.1007/11811305_3

[53] Baudat G., and Anouar F., “Generalized Discriminant Analysis Using a Kernel Approach”, *Neural Computation*, 2000, vol. 12, iss. 10, pp. 2385–2404. doi: https://doi.org/10.1162/089976600300014980

[54] S. Benndorf, G. Kakulapati, A. Pham and others (2015), “Fighting Mobile Fraud in the Programmatic Era*”, AppLift GmbH*, 2015.

[55] Fraudwatch. [Online]. Available: http://www.fraudshields.com/

[56] Andrii Yarovyi, Raisa Ilchenko, Ihor Arseniuk, Yevhene Shemet, Andrzej Kotyra, and Saule Smailova, "An intelligent system of neural networking recognition of multicolor spot images of laser beam profile", in *Proc. SPIE 10808, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics*

Experiments 2018, 108081B, 2018. doi: https://doi.org/10.1117/12.2501691

[57] V. Kozhemyako, L. Timchenko, and A. Yarovyy, "Methodological Principles of Pyramidal and Parallel-Hierarchical Image Processing on the Base of Neural-Like Network Systems", *Advances in Electrical and Computer Engineering*, vol.8, no.2, pp. 54-60, 2008. doi:10.4316/AECE.2008.02010

[58] M. Granik, V. Mesyura, and A. Yarovyi, "Determining Fake Statements Made by Public Figures by Means of Artificial Intelligence", in *Proc. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, Lviv, 2018, pp. 424-427. doi: 10.1109/STC-CSIT. 2018.8526631

[59] Анализ клиентских баз данных. Выявление мошенничества (fraud detection) на базе STATISTICA Data Miner. [Online]. Available: http://statsoft.ru/solutions/ExamplesBase/branches/detail.php?ELEMENT_ID=834

[60] А.А. Яровий, та Т.Д. Польгуль, “Підвищення продуктивності обчислювальних процесів в паралельно-ієрархічній мережі за допомогою Framework Benchmark Akka”, Збірник тез доповіді на *VII Міжнародній науково-технічній конференції «Фотоніка ОДС-2015»*, Вінниця, 2015, с. 9.

[61] Д.П. Польгуль, та Т.Д. Польгуль, “Концептуальна модель інформаційної безпеки”, на *Міжнародній науково-практичній конференції «Тенденції управління фінансовими та інноваційними процесами в умовах ринкових перетворень», секція «Актуальні проблеми розвитку управління сучасним підприємством»,* Вінниця, ВНТУ, 2012, с. 340-342.

[62] Т.Д. Польгуль, “Порівняльний аналіз Apache Spark та Apache Flink для роботи з Big Data”, на *XLV науково-технічній конференції підрозділів ВНТУ,* Вінниця, 2016. [Електронний ресурс]. Режим доступу: https://ir.lib.vntu.edu.ua/handle/123456789/11619

[63] А.Г. Кюльян, Т.Д. Польгуль, та М.Б. Хазін, “Математична модель рекомендаційного сервісу на основі методу колаборативної фільтрації”, *Комп’ютерні технології та Інтернет в інформаційному суспільстві*, с. 226–227, 2012. Режим доступу: http://ir.lib.vntu.edu.ua/bitstream/handle/

123456789/7911/226227.pdf?sequence=1&isAllowed=y

[64] T. Joachims, “Training linear SVMs in linear time”, in *Proc. 12th ACM SIGKDD international conference on Knowledge discovery and data mining KDD ’06, ACM*, New York, USA, 2006, pp. 217–226.

[65] J. Platt, *Probabilistic outputs for support vector machines and comparison to regularized likelihood methods*. MIT Press, 2000, pp. 61–74.

[66] J. L. Bentley, 1975. “Multidimensional binary search trees used for associative searching”, *Communications of the ACM*, vol. 18, iss. 9, pp. 509–517, 1975.

[67] N. Roussopoulos, S. Kelley, and F. Vincent, “Nearest neighbor queries”, in *Proc. ACM-SIGMOD International Conference on Management of Data*, Місто, 1995.

[68] Метод k-ближайших соседей. [Электронный ресурс]. Режим доступа: http://om.univ.kiev.ua/users_upload/15/upload/file/pr_lecture_03.pdf

[69] К. Воронцов, “Машинное обучение (курс лекций)”. [Электронный ресурс]. Режим доступа: http://www.machinelearning.ru/wiki/index.php?title =Машинное_обучение_(курс_лекций%2C_К.В.Во ронцов)

[70] К.Д. Маннинг, П. Рагхаван, и Х. Шютце, *Введение в информационный поиск*. Москва, Россия: Вильямс, 2011.

[71] J. A. Hartigan, and M. A. Wong, “A k-means clustering algorithm*”, Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no.1 (1979), pp. 100–108.

[72] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, “A geometric framework for unsupervised anomaly detection”, in *Proc. Applications of Data Mining in Computer Security*. New York, USA: Kluwer Academics, 2002, pp. 78–100.

[73] Jain A. K., *Data Clustering: A Review*. Michigan State University. [Online]. Available: E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, 2002. “A geometric framework for unsupervised anomaly detection”, in *Proc. Applications of Data Mining in Computer Security*. Kluwer Academic Publishers, 2002, pp. 77–101.

[74] Jain, A. K. and Dubes, R. C. *Algorithms for Clustering Data*. New Jersey, USA: Prentice Hall, 1988.

[75] Deep Neural Network. [Online]. Available: https://www.techopedia.com/ definition/32902/deep-neural-network

[76] Training Deep Neural Network. [Online]. Available: https://www.techopedia.com/definition/32902/deep-neural-network

[77] Е. Е. Пятикоп, “Исследованине метода коллаборативной фильтрации на основе сходства элементов”. *Наукові праці Донецького національного технічного університету. Серія “Інформатика, кібернетика та обчислювальна техніка”*, № 2 (18), с. 109-114, 2013.

[78] Э. Беккенбах, и Р. Беллман, *Введение в неравенства*. Москва: Мир, 1965.

[79] Р. А. Шмойлова и др., *Теория статистики: Учебник*. Москва, Россия: Финансы и Статистика, 2001.

[80] J. L. Rodgers, and W. A. Nicewander, “Thirteen ways to look at the correlation coefficient”, *The American Statistician*, vol. 42, issue 1, pp. 59-66, 2012.

[81] Ф. Дж. Мак-Вильямс, и Н. Дж. А. Слоэн, *Теория кодов, исправляющих ошибки.* Москва: Связь, 1979.

[82] В. Д. Дмитриенко, и А. Ю. Заковоротный, “Нейронная сеть, использующая расстояние Хемминга, для распознавания изображений на границах нескольких классов”, *Вісник Національного технічного університету “ХПІ”. Інформатика та моделювання*, № 39, с. 57-67, 2013.

[83] Р. Блейхут, *Теория и практика кодов, контролирующих ошибки = Theory and Practice of Error Control Codes*. Москва: Мир, 1986.

[84] *Методичні вказівки до виконання комп’ютерних практикумів з навчальної дисципліни «Інтелектуальний аналіз даних».* Київ, Україна: КПІ ім.Ігоря Сікорського, 2017.

[85] И. И. Елисеева, и В. О. Рукавишников, *Группировка, корреляция, распознавание образов: (статистические методы классификации и измерения связей)*. Москва: Статистика, 1977.

[86] С.І. Альперт, “Основні міри подібності та нові підходи до їх застосування при класифікуванні гіперспектральних космічних зображень”, *Математичні машини і системи*, № 1, 2019.

[87] К.Д. Маннинг, П. Рагхаван, и Х. Шютце, *Введение в информационный поиск*. Москва, Россия: Вильямс, 2011.

[88] Т. О. Говорущенко, О. С. Савенко, С. М. Лисенко, "Забезпечення кібербезпеки комп’ютерних систем в умовах інноваційного розвитку України", *Матеріали Всеукраїнської науково-практичної конференції «Безпека соціально-економічних процесів в кіберпросторі»*, Київ, 27 березня 2019 р., с. 41-42.

[89] Что такое Big data: собрали всё самое важное о больших данных. [Электронный ресурс]. Режим доступа: https://rb.ru/howto/chto-takoe-big-data/

[90] Леоненков А.В., *Самоучитель UML.* Санкт-Петербург, Россия: БХВ-Петербург, 2002. ISBN 5-94157-878-4, 978-5-94157-878-8

[91] A. Dennis, B. Haley, and D. Tegarden, *Systems Analysis and Design with UML* . USA: Wiley, 2012. ISBN 978-1118037423

[92] А. А. Яровий, та Т. Д. Польгуль, Комп'ютерна програма «Програмний модуль збору даних інформаційної технології виявлення шахрайства при інсталюванні програмних додатків». *Свідоцтво про реєстрацію авторського права на твір № 76348.* Київ: Міністерство економічного розвитку і торгівлі України, 2018.

[93] А. А. Яровий, та Т. Д. Польгуль, Комп'ютерна програма «Програмний модуль визначення схожості користувачів інформаційної технології виявлення шахрайства при інсталюванні програмних додатків». *Свідоцтво про реєстрацію авторського права на твір № 76347*. Київ: Міністерство економічного розвитку і торгівлі України, 2018.

[94] T. Polhul, “Fraud detection classifier”, Github. [Online]. Available: https://github.com/tanapolg/fraud-detection-classifier

[95] Kaggle. [Online]. Available: https://www.kaggle.com/

[96] R. Johnson, *Applied Multivariate Statistical Analysis*. New Jersey, USA: Prentice Hall, 1992.

[97] Оценивание и сравнение нейросетевых моделей. [Электронный ресурс]. Режим доступа: https://helpiks.org/6-10672.html

[98] Gamification by University of Pennsilvania. [Online]. Available: https://www.coursera.org/learn/gamification

[99] G. Baudat, and F. Anouar, “Generalized Discriminant Analysis Using a Kernel

Approach”, *Neural Computation*, vol. 12, no. 10, pp. 2385–2404, 2000. doi: https://doi.org/10.1162/089976600300014980

[100] Instagram. URL: https://www.instagram.com/

[101] SocialKit. Популярная программа для Instagram. [Online]. Available: https://socialkit.ru/

[102] В. М. Дубовой, та О. Д. Никитенко, *Оптимізація підсистем збору даних АСУТП в умовах комбінованої невизначеності*. *Монографія.* Вінниця, Україна: УНІВЕРСУМ-Вінниця, 2011. ISBN 978-966-641-434-5

[103] Ш. Эхтер, и Д. Робертс, *Многоядерное программирование*. Санкт-Петербург, Россия: Питер, 2010. ISBN 978-5-388-00091-0

[104] Mark D. Hill, and Michael R. Marty, “Amdahl's Law in the Multicore Era”, *IEEE Computer*, vol. 41, issue 7, pp. 33-38, 2008.

[105] S.L. Blumin, I.A. Shuykova, P.V. Sarajev, and I.V. Cherpakov, *Fuzzy Logic: Algebraic Foundations and Applications*. *Monograph* . Lipetsk, Russia: LESI, 2002.

[106] L. Zade, *The concept of a linguistic variable and its application to making approximate decisions*. Москва: Mir, 1976.

[107] В. І. Месюра, та Л. М. Ваховська, *Методичні вказівки до виконання курсового проекту з дисципліни "Системи прийняття рішень з нечіткою логікою" для студентів спеціальності "Інтелектуальні системи прийняття рішень"*. Вінниця, Україна: ВНТУ, 2005.

[108] T.D. Polhul, A.A. Yarovyi, R.Romaniuk, P.Komada, N. Askarova "Method of data anomaly detection in the process of mobile applications installation", Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019, 111761Y; https://doi.org/10.1117/ 12.2536855

ДОДАТКИ

Додаток А Документи щодо впровадження результатів роботи

"ЗАТВЕРДЖУЮ"
Заступник директора Товариства
з обмеженою відповідальністю
"ВІН ІНТЕРАКТИВ"
Томашпольський О.С.

"31" жовтня 2019 р.

АКТ
про впровадження результатів
дисертаційної роботи Польгуль Тетяни Дмитрівни

Комісія у складі: голови комісії заступника директора Томашпольського Олександра Самойловича, заступника директора Вінничука Олександра Петровича, аналітика з питань фінансово-економічної безпеки Нухимовича Олега Адольфовича,

склали цей акт про те, що результати дисертаційної роботи Польгуль Тетяни Дмитрівни отримали практичну реалізацію у вигляді впровадження створеного в роботі програмного забезпечення для виявлення шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних на основі програмних продуктів, розроблених в рамках договору про творчу співдружність номер реєстрації 011019/1.

Комісія підтверджує, що використані наступні результати дисертаційної роботи:

1. Алгоритми процесу подолання різнорідності даних;
2. Моделі процесу подолання різнорідності даних;
3. Модель класифікації вхідних даних для виявлення аномалій в Big Data;
4. Методика проведення експериментальних досліджень шахрайства при інсталюванні мобільних додатків.

Ефект від впровадження зазначених результатів полягає у підвищенні точності визначення шахрайських користувачів мобільних додатків, в наслідок чого витрати на маркетингові кампанії було знижено.

Голова комісії Томашпольський О.С.

Члени комісії: Вінничук О.П.

Нухимович О.А.

IMPLEMENTATION CERTIFICATE

Garuda AI B. V., hereby certifies the completion of all operational implementation activities by Tetiana Polhul. A series of subsystems and algorithms were designed, tested and delivered to Garuda AI B. V. The public algotithms are discribed in "Polhul T., Yarovyi A. Method of Fraudster Fingerprint Formation During Mobile Application Installations // Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Metz, France, September 18-21, 2019, Volume 1, p. 1099–1103", and the code was delivered to Garuda AI B.V. under public Apache License 2.0: https://github.com/tanapolg/fraud-detection-classifier. These subsystems and algorithms are intended to be the basis of automated website for fraud and hacker attacks detection (FDHA). The FDHA subsystem operates as a key part of fraud and hacker attack detection system provided by Garuda AI B. V.

Developed software module includes:

1) Component that includes a series of subsystems and algorithms related to available data analysis; intellectual processing of available data; developing a database and knowledge base (for detecting fraudsters); classification model building and user classification; users' templates formation; the generalized fraudsters fingerprint formation. The developed information technology allows the processing of various input data, which in the process gives the opportunity to form a generalized template of a fraudster.
2) Fraud and hacker attack detection component that allows identifying organic users and hackers (fraudsters).
3) Administrative interface for the developed module to allow general subsystem control and maintenance.
4) User interface for the developed module.

The information technology for fraud and hacker attack detection based on methods, models and algorithms for detecting fraud proposed by Tetiana Polhul is a key component of the fraud detection engine and user class detecting is a key part of user class detecting subsystem.

Signature
Viktor Radchenko

22 September 2019
Date

Garuda AI B. V.
Het Poortgebouw,
Beech Avenue 54-62,
Schiphol, 1119 PW,
Netherland

"ЗАТВЕРДЖУЮ"

Директор Товариства з обмеженою
відповідальністю 4ХАЙТЕК
Дмитриева Татьяна Юрьевна

"31" жовтня 2019 р.

АКТ
про впровадження результатів
дисертаційної роботи Польгуль Тетяни Дмитрівни

Комісія у складі: Директора ТОВ 4ХАЙТЕК Дмитрієвої Тетяни Юріївни, техніка з комп'ютерних систем Павлова Дмитра Анатолійовича, аналітика операційного та прикладного програмного забезпечення Нухимовича Ігоря Олеговича,

склали цей акт про те, що матеріали дисертаційної роботи на здобуття наукового ступеня кандидата технічних наук Польгуль Т. Д. отримали практичну реалізацію у вигляді впровадження створеного в роботі програмного забезпечення для виявлення шахрайства на основі програмних продуктів, розроблених в рамках договору про творчу співдружність номер реєстрації 011019/2.

Комісія підтверджує, що використані наступні результати дисертаційної роботи:

1. Узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків.
2. Алгоритм виявлення шахрая при інсталюванні мобільних додатків;
3. Алгоритм створення узагальненого портрету шахрая;
4. Методика проведення експериментальних досліджень шахрайства при інсталюванні мобільних додатків.

Ефект від впровадження зазначених результатів полягає у підвищенні точності та швидкодії визначення шахраїв, в наслідок чого було знижено витрати компанії.

Голова комісії Дмитрієва Т.Ю.

Члени комісії Павлов Д.А.

Нухимович І.О.

ЗАТВЕРДЖУЮ
Директор ПП «Літсофт»

_____ О. О. Летичевський
«__» вересня 2019 р.

АКТ
про впровадження результатів
дисертаційної роботи Польгуль Тетяни Дмитрівни

Даним актом засвідчується, що в ПП «Літсофт» використано матеріали дисертаційної роботи на здобуття наукового ступеня кандидата технічних наук Польгуль Т.Д.:

1. Алгоритм подолання різнорідності даних;
2. Узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків;
3. Алгоритм створення узагальненого портрету шахрая.

Ефект від впровадження зазначених результатів полягає у зниженні витрат на маркетингові кампанії, які займаються шахрайством завдяки попередньому їх виявленню з використанням розробленого програмного забезпечення.

Відповідальні за впровадження:

Директор ПП «Літсофт» *О.О. Летичевський*

Бухгалтер *В.В.Марциновська*

ЗАТВЕРДЖУЮ
Перший проректор з науково-педагогічної роботи
по організації навчального процесу та його
науково-методичного забезпечення Вінницького
національного технічного університету

О. М. Василевський
"30" жовтня 2019 р.

АКТ
про впровадження результатів дисертаційної роботи
Польгуль Тетяни Дмитрівни

Члени комісії у складі к.т.н., професора, директора Головного центру організаційного та методичного забезпечення навчання Лисенко Г. Л., декана ФІТКІ, д.т.н., професора Азарова О. Д., завідувача кафедри КН, д.т.н., професора Ярового А. А. склали цей акт про те, що у 2016-2019 роках на кафедрі комп'ютерних наук Вінницького національного технічного університету впроваджено у навчальний процес результати дисертаційної роботи на здобуття наукового ступеня кандидата технічних наук, отримані Польгуль Т. Д., а саме:

– інформаційна технологія виявлення шахрайства при інсталюванні мобільних додатків;

– узагальнений метод виявлення шахрайства при інсталюванні мобільних додатків;

– моделі та алгоритми процесу подолання різнорідності вхідних даних.

Теоретичні та практичні результати дисертаційної роботи використано в навчальному процесі на кафедрі комп'ютерних наук ВНТУ при підготовці фахівців зі спеціальності 122 – Комп'ютерні науки та інформаційні технології (ОПП "Комп'ютерні науки"), спеціальності 122 – Комп'ютерні науки (ОПП "Комп'ютерні науки") – І (бакалаврського) рівня вищої освіти, а також спеціальності 122 – Комп'ютерні науки (ОПП "Системи штучного інтелекту") – ІІ (магістерського) рівня вищої освіти при викладанні таких дисциплін, як: «Інтелектуальний аналіз даних», «Методи та системи штучного інтелекту», «Проектування інформаційних систем», «Технології розподілених систем та паралельних обчислень», «Сучасні інформаційні технології в науці та освіті», що дозволило підвищити рівень практичних навиків при застосуванні інтелектуальних інформаційних технологій.

Директор Головного центру
організаційного та методичного
забезпечення навчання — Г. Л. Лисенко

Декан ФІТКІ — О. Д. Азаров

Завідувач кафедри КН — А. А. Яровий

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
вул. 40 років Жовтня, 27, м. Херсон, 73003. Тел.: +38(0552) 32-67-05, 32-67-31; факс 49-21-14; e-mail: office@ksu.ks.ua; http://www.kspu.edu
МФО 820172 код за ЄДРПОУ 02125609 р/р 3522 7222 000120; 3521 2022 000120 банк Держказначейська служба України, м. Київ

01.11 2017 р. № 15-50/1538

На № _____ від _____ 201__ р.

ДОВІДКА

про апробацію і впровадження дисертаційного дослідження Польгуль Тетяни Дмитрівни на здобуття наукового ступеня кандидата технічних наук зі спеціальності 05.13.06 - Інформаційні технології

Протягом 2017 - 2019 років Т. Д. Польгуль на базі кафедри інформатики, програмної інженерії та економічної кібернетики Херсонського державного університету проводила апробацію та впровадження результатів дисертаційного дослідження інформаційної технології виявлення шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних.

У процесі співпраці зі студентами, магістрантами, аспірантами кафедри інформатики, програмної інженерії та економічної кібернетики з апробованої проблематики обговорювались ідеї здобувачки стосовно можливостей застосування окремих положень дисертації, що стосуються машинного навчання та інтелектуального аналізу даних у процесі підготовки майбутніх фахівців. Матеріали дисертаційного дослідження розглядалися на засіданнях наукових семінарів кафедри та отримали позитивні відгуки.

Основні положення та висновки дисертаційного дослідження Польгуль Т.Д., використовувалися при викладанні дисциплін “Безпека програм та даних”, “Інтелектуальний аналіз даних” у процесі проведення лекційних та практичних занять.

Довідка про апробацію і впровадження дисертаційного дослідження Польгуль Тетяни Дмитрівни на тему “Інформаційна технологія виявлення шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних” обговорена і затверджена на засіданні кафедри інформатики, програмної інженерії та економічної кібернетики (протокол № 3 від 31.10.2019 р.).

Проректор з наукової роботи Сергій ОМЕЛЬЧУК

Песчаненко В.С.
(0552)-32-67-68

Додаток Б Основні лістинги програмного забезпечення

```
import pandas as pd
import numpy as np

# Import and store dataset
credit_card_data = pd.read_csv('creditcard.csv')
#print(credit_card_data)

#my
class_names = credit_card_data['Class']
# To separate our data into few sets (training etc):
# Splitting data into 4 sets:
# 1. Shuffle/randomize data
# 2. One-hot encoding
# 3. Normalize our data (we'll convert negative data
# into numbers between 0 and 1)
# 4. Splitting up X/y values (x values - our inputs,
# y values - out outputs; x - columns from 1 till 28 in our .csv,
# y - one-hot encodings - 0..1 - will tell us our transactions are fraudulent or not)
# 5. Convert data_frames (creadit_card_data from .csv)
# to numpy arrays (float32 - TensorFlow loves to work with float32)
# 6. Splitting the final data into X/y train/test - so we'll have 4 arrays:
# X train, y train, X test, y test

# Shuffle and randomize data
shuffled_data = credit_card_data.sample(frac=1)
# Change Class column into Class_0 ([1 0] for legit data) and Class_1 ([0 1] for fraudulent data)
one_hot_data = pd.get_dummies(shuffled_data, columns=['Class'])
# Change all values into numbers between 0 and 1
normalized_data = (one_hot_data - one_hot_data.min()) / (one_hot_data.max() - one_hot_data.min())
# Store just columns V1 through V28 in df_X and columns Class_0 and Class_1 in df_y
df_X = normalized_data.drop(['Class_0', 'Class_1'], axis=1)
df_y = normalized_data[['Class_0', 'Class_1']]
# Convert both data_frames into np arrays of float32
ar_X, ar_y = np.asarray(df_X.values, dtype='float32'), np.asarray(df_y.values, dtype='float32')
# Allocate first 80% of data into training data and remaining 20% into testing data
train_size = int(0.8 * len(ar_X))
(raw_X_train, raw_y_train) = (ar_X[:train_size], ar_y[:train_size])
(raw_X_test, raw_y_test) = (ar_X[train_size:], ar_y[train_size:])

# Gets a percent of fraud vs legit transactions (0.0017% of transactions are fraudulent)
count_legit, count_fraud = np.unique(credit_card_data['Class'], return_counts=True)[1]
fraud_ratio = float(count_fraud / (count_legit + count_fraud))
print('Percent of fraudulent transactions: ', fraud_ratio)

# Applies a logit weighting of 578 (1/0.0017) to fraudulent transactions to cause model to pay more attention to them
weighting = 1 / fraud_ratio
raw_y_train[:, 1] = raw_y_train[:, 1] * weighting

import tensorflow as tf

# 30 cells for the input
input_dimensions = ar_X.shape[1]
# 2 cells for the output
output_dimensions = ar_y.shape[1]
# 100 cells for the 1st layer
num_layer_1_cells = 100
# 150 cells for the second layer
num_layer_2_cells = 150

# We will use these as inputs to the model when it comes time to train it (assign values at run time)
X_train_node = tf.placeholder(tf.float32, [None, input_dimensions], name='X_train')
y_train_node = tf.placeholder(tf.float32, [None, output_dimensions], name='y_train')

# We will use these as inputs to the model once it comes time to test it
X_test_node = tf.constant(raw_X_test, name='X_test')
y_test_node = tf.constant(raw_y_test, name='y_test')
```

```
# First layer takes in input and passes output to 2nd layer
weight_1_node = tf.Variable(tf.zeros([input_dimensions, num_layer_1_cells]), name='weight_1')
biases_1_node = tf.Variable(tf.zeros([num_layer_1_cells]), name='biases_1')

# Second layer takes in input from 1st layer and passes output to 3rd layer
weight_2_node = tf.Variable(tf.zeros([num_layer_1_cells, num_layer_2_cells]), name='weight_2')
biases_2_node = tf.Variable(tf.zeros([num_layer_2_cells]), name='biases_2')

# Third layer takes in input from 2nd layer and outputs [1 0] or [0 1] depending on fraud vs legit
weight_3_node = tf.Variable(tf.zeros([num_layer_2_cells, output_dimensions]), name='weight_3')
biases_3_node = tf.Variable(tf.zeros([output_dimensions]), name='biases_3')


# Function to run an input tensor through the 3 layers and output a tensor that will give us a fraud/legit result
# Each layer uses a different function to fit lines through the data and predict whether a given input tensor will \
#   result in a fraudulent or legitimate transaction
def network(input_tensor):
    # Sigmoid fits modified data well
    layer1 = tf.nn.sigmoid(tf.matmul(input_tensor, weight_1_node) + biases_1_node)
    # Dropout prevents model from becoming lazy and over confident
    layer2 = tf.nn.dropout(tf.nn.sigmoid(tf.matmul(layer1, weight_2_node) + biases_2_node), 0.85)
    # Softmax works very well with one hot encoding which is how results are outputted
    layer3 = tf.nn.softmax(tf.matmul(layer2, weight_3_node) + biases_3_node)
    return layer3


# Used to predict what results will be given training or testing input data
# Remember, X_train_node is just a placeholder for now. We will enter values at run time
y_train_prediction = network(X_train_node)
y_test_prediction = network(X_test_node)

# Cross entropy loss function measures differences between actual output and predicted output
cross_entropy = tf.losses.softmax_cross_entropy(y_train_node, y_train_prediction)

# Adam optimizer function will try to minimize loss (cross_entropy) but changing the 3 layers' variable values at a
#   learning rate of 0.005
optimizer = tf.train.AdamOptimizer(0.005).minimize(cross_entropy)

import itertools
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
```

```
            plt.text(j, i, format(cm[i, j], fmt),
                     horizontalalignment="center",
                     color="white" if cm[i, j] > thresh else "black")

        plt.ylabel('True label')
        plt.xlabel('Predicted label')
        plt.tight_layout()


from sklearn.metrics import confusion_matrix
from sklearn import metrics
from ggplot import *
# Function to calculate the accuracy of the actual result vs the predicted result
def calculate_accuracy(actual, predicted, buildMatrix):
    actual = np.argmax(actual, 1)
    predicted = np.argmax(predicted, 1)
    # my confusion matrix
    if buildMatrix:
        a = 0
        f = 0
        for i in actual:
            if actual[i] == 0:
                a = a + 1
            if actual[i] == 1:
                f = f + 1
        print('actual: ')
        print(actual.size)
        print(a)
        print(f)
        ap = 0
        fp = 0
        for i in predicted:
            if predicted[i] == 0:
                ap = ap + 1
            else:
                fp = fp + 1
        print('predicted: ')
        print(predicted.size)
        print(ap)
        print(fp)

        fpr, tpr, threshold = metrics.roc_curve(actual, predicted)
        roc_auc = metrics.auc(fpr, tpr)

        # method I: plt
        plt.title('Receiver Operating Characteristic')
        plt.plot(fpr, tpr, 'b', label='AUC = %0.2f' % roc_auc)
        plt.legend(loc='lower right')
        plt.plot([0, 1], [0, 1], 'r--')
        plt.xlim([0, 1])
        plt.ylim([0, 1])
        plt.ylabel('True Positive Rate')
        plt.xlabel('False Positive Rate')
        plt.show()

        # method II: ggplot
        df = pd.DataFrame(dict(fpr=fpr, tpr=tpr))
        ggplot(df, aes(x='fpr', y='tpr')) + geom_line() + geom_abline(linetype='dashed')
        # Compute confusion matrix
        # cnf_matrix = confusion_matrix(actual, predicted)
        # np.set_printoptions(precision=2)

        # Plot non-normalized confusion matrix
        # plt.figure()
        # plot_confusion_matrix(cnf_matrix, classes=class_names,
        #                       title='Confusion matrix, without normalization')

        # Plot normalized confusion matrix
        # plt.figure()
        # plot_confusion_matrix(cnf_matrix, classes=class_names, normalize=True,
```

```
        #                   title='Normalized confusion matrix')
        # plt.interactive(False)
        # plt.show(block=True)
    return (100 * np.sum(np.equal(predicted, actual)) / predicted.shape[0])

num_epochs = 100

import time

with tf.Session() as session:
    tf.global_variables_initializer().run()
    for epoch in range(num_epochs):

        start_time = time.time()

        _, cross_entropy_score = session.run([optimizer, cross_entropy],
                                             feed_dict={X_train_node: raw_X_train, y_train_node: raw_y_train})

        if epoch % 10 == 0:
            timer = time.time() - start_time

            print('Epoch: {}'.format(epoch), 'Current loss: {0:.4f}'.format(cross_entropy_score),
                  'Elapsed time: {0:.2f} seconds'.format(timer))

            final_y_test = y_test_node.eval()
            final_y_test_prediction = y_test_prediction.eval()
            final_accuracy = calculate_accuracy(final_y_test, final_y_test_prediction, False)
            print("Current accuracy: {0:.2f}%".format(final_accuracy))

    final_y_test = y_test_node.eval()
    final_y_test_prediction = y_test_prediction.eval()
    final_accuracy = calculate_accuracy(final_y_test, final_y_test_prediction, False)
    print("Final accuracy: {0:.2f}%".format(final_accuracy))

final_fraud_y_test = final_y_test[final_y_test[:, 1] == 1]
final_fraud_y_test_prediction = final_y_test_prediction[final_y_test[:, 1] == 1]
final_fraud_accuracy = calculate_accuracy(final_fraud_y_test, final_fraud_y_test_prediction, True)
print('Final fraud specific accuracy: {0:.2f}%'.format(final_fraud_accuracy))
```

Додаток В Список експертів, які проводили оцінювання

Деякі коефіцієнти було визначено на основі експертного оцінювання. Більшість експертів давали свою думку щодо розробленої інформаційної технології. Експертами виступили ведучі програмісти IT-компаній:

1. Грузман Давид, CEO and Founder at Nestlogic.
2. Летичевський Олександр, провідний науковий співробітник інституту кібернетики ім. В.М. Глушкова НАНУ, Chief Research Office at Garuda AI B.V., Netherlands.
3. Песчаненко В. С., зав. каф. інформатики, доктор ф. м. наук Херсонського університету, Chief Software Architect at Garuda AI B.V., Netherlands.
4. Vladyslav Lysychkin, Software Engineer at Google, Zurich, Switzerland. Working on natural language understanding for Google Assistant.
5. Москвін Олексій Михайлович, CEO/CTO at Plexteq, PhD degree in Information Technology area.
6. Стах Олексій, Software Architect at Microsoft, Seattle, USA.
7. Каменских Антон, кандидат технічних наук, доцент кафедри "Автоматика і телемеханіка", куратор освітньої програми "Інформаційна безпека автоматизованих систем", спеціалізація "Безпека відкритих інформаційних систем" набір до 2019 р. – Забезпечення інформаційної безпеки розподілених інформаційних систем, ФГБОУ ВО Пермський державний технічний університет.
8. Кухар Роман, Software Engineer, Sandbox.
9. Ковальчук Олександр, Senior Software Engineer, Plexteq.
10. Trenton Miller, Senior Technical Recruiter at Redfin, Seattle, USA.
11. Ліщишин Володимир, R&D Technical Leader, AUG.global.
12. Ліщенко Тарас, Computer Vision Engineer.
13. Лущиков Юрій, Project Manager, Onseo.

Додаток Д Сертифікати на публікації та виступи

UNIVERSITY OF CAMBRIDGE

CAMBRIDGE BIG DATA

IRDTA BRUSSELS - LONDON

BigDat 2019

5th International Winter School on Big Data

January 7-11, 2019

Cambridge, United Kingdom

THIS IS TO CERTIFY THAT:

Tetiana Polhul has delivered a 5-minute presentation with the title ***Specifics of Intelligent System Development for Detecting Mobile App Install Fraud***.

Cambridge, January 11, 2019

Carlos Martín-Vide
Professor, Rovira i Virgili University

UNIVERSITY OF CAMBRIDGE

CAMBRIDGE BIG DATA

IRDTA BRUSSELS - LONDON

BigDat 2019

5th International Winter School on Big Data

January 7-11, 2019

Cambridge, United Kingdom

THIS IS TO CERTIFY THAT:

Tetiana Polhul has successfully participated in BigDat 2019, which consisted of 39 hours of lectures.

Cambridge, January 11, 2019

Filippo Spiga
Cambridge Big Data Initiative

Carlos Martín-Vide
Professor, Rovira i Virgili University

INSTITUTE OF RESEARCH ENGINEERS AND SCIENTISTS

International Conference on
Innovative Engineering Technologies

Certificate

This is to certify that Tetiana Polhul has presented a paper entitled "Development of an Intelligent System for Detecting Mobile App Install Fraud" at the International Conference on Innovative Engineering Technologies (ICIET) held in Bangkok, Thailand on 21st-22nd March, 2019.

IRES-ICIETBNGK-21039-10094

Paper ID

Chairman
INSTITUTE OF RESEARCH ENGINEERS AND SCIENTISTS

THIS AWARD CERTIFICATE

CERTIFIES THAT

Tetiana Polhul

submitted THE BEST PAPER to 10th International Scientific and Technical Conference

"Dependable Systems, Services and Technologies"

DESSERT 2019

organized by

Leeds Beckett University, Leeds, United Kingdom

National Aerospace University KhAI, Kharkiv, Ukraine

IEEE United Kingdom and Ireland Section

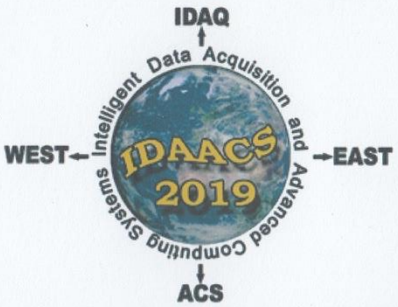
IEEE Ukraine Section

United Kingdom, Leeds, LBU, 5-7th of June, 2019

General Chair — *Prof. Vyacheslav Kharchenko*

PC Chair — *Prof. Hissam Tawfik*


IEEE
INTERNATIONAL SCIENTIFIC TECHNICAL CONFERENCE
IEEE DeSSerT 2019 UNITED KINGDOM LEEDS JUNE 5 -7
DEPENDABLE SYSTEMS, SERVICES & TECHNOLOGIES
THIS CERTIFICATE
CERTIFIES THAT
Tetiana Polhul
is the SPEAKER of 10th International Scientific and Technical Conference
"Dependable Systems, Services and Technologies"
DESSERT 2019
organized by
Leeds Beckett University, Leeds, United Kingdom
National Aerospace University KhAI, Kharkiv, Ukraine
IEEE United Kingdom and Ireland Section
IEEE Ukraine Section
United Kingdom, Leeds, LBU, 5-7th of June, 2019
General Chair
Prof. Vyacheslav Kharchenko
PC Chair
Prof. Hissam Tawfik
LBU



Certificate of Attendance

IDAACS'2019

The IEEE 10 th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications

ENIM (Ecole Nationale d'Ingénieurs de Metz) and LCOMS (Laboratory of Conception, Optimisation and Modelling of Systems), University of Lorraine, France, September 18-21, 2019

Tetiana Polhul

Kondo H. Adjallah,
IDAACS'2019 Co-Chairman

Anatoliy Sachenko,
IDAACS'2019 Co-Chairman


НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
1885
ХПІ
СЕРТИФІКАТ
Польщук Тетяна Дмитрівна
виступила з доповіддю на V-й Міжнародній науково-технічній конференції студентів, магістрів та аспірантів
«ІНФОРМАТИКА, УПРАВЛІННЯ ТА ШТУЧНИЙ ІНТЕЛЕКТ»
Заступник голови конференції,
Вчений секретар НТУ «ХПІ»
д.т.н., професор
Заковоротний О.Ю.
20-22 листопада 2018 р.
DATE
Харків
LOCATION

Додаток Е Список публікацій за темою дисертації

[1] T. Polhul, and A. Yarovyi, “Development of a method for fraud detection in heterogeneous data during installation of mobile applications”, *Eastern-European Journal of Enterprise Technologies*, no. 1/2 (97), 2019. doi: 10.15587/1729-4061.2019.155060

[2] A. Yarovyi, and T. Polhul, “Applied Aspects of Implementation of Intelligent Information Technology for Fraud Detection During Mobile Applications Installation”, *Advances in Intelligent Systems and Computing IV. CCSIT 2019: Advances in Intelligent Systems and Computing, Springer*, Cham, Switzerland, vol 1080, pp. 377-386, 2019. doi: https://doi.org/10.1007/978-3-030-33695-0_26

[3] T. Polhul, “Conceptual Model of an Intelligent System for Detecting Fraud During Mobile Applications Installation”, in *Proc. 10th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Leeds, United Kingdom, pp. 167-174, 2019. doi: 10.1109/DESSERT.2019.8770030. Режим доступу: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8770030&isnumber=8770005.

[4] T.D. Polhul, A.A. Yarovyi, R.Romaniuk, P.Komada, N. Askarova "Method of data anomaly detection in the process of mobile applications installation", *Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019*, 111761Y; https://doi.org/10.1117/ 12.2536855

[5] T. Polhul and A. Yarovyi, "Method of Fraudster Fingerprint Formation During Mobile Application Installations", *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Metz, France, 2019, pp. 1099-1103. doi: 10.1109/IDAACS.2019.8924369. Режим доступу: https://ieeexplore.ieee.org/document/8924369.

[6] A. Yarovyi, and T. Polhul, “Intelligent information technology for fraud detection during mobile applications installation”, in *Proc. 14th International conference "Computer sciences and Information technologies" (CSIT 2019)*, pp. 1-5,

Lviv, 2019. doi: 10.1109/STC-CSIT.2019.8929827. Режим доступу: https://ieeexplore.ieee.org/ document/8929827.

[7] T. Polhul, “Development of an intelligent system for detecting mobile app install fraud”, *International Journal of Advances in Electronics and Computer Science (IJAECS),* vol. 6, no. 7, pp. 13-17, July, 2019.

[8] А.А. Яровий, О.Н. Романюк, І.Р. Арсенюк, та Т.Д. Польгуль, “Виявлення шахрайства при інсталюванні програмних додатків з використанням інтелектуального аналізу даних”, *Наукові праці Донецького національного технічного університету. Серія “Інформатика, кібернетика та обчислювальна техніка”*, № 2 (25), с. 126-131, 2017. Режим доступу: http://science.donntu.edu.ua/ wp-content/uploads/2018/03/ikvt_2017_2_site-1.pdf

[9] Т. Д. Польгуль, та А. А. Яровий, “Метод подолання різнорідності даних для виявлення шахрайства при інсталюванні мобільних додатків”, *Вісник СНУ ім. В. Даля – Сєвєродонецьк: СНУ ім. В. Даля*, № 7 (248), с.60-69, 2018.

[10] Т.Д. Польгуль, та А.А. Яровий, “Аналіз різнорідних даних в інтелектуальних системах виявлення шахрайства”, *Вісник Вінницького політехнічного інституту*, № 2, с. 78-90, 2019.

[11] Т.Д. Польгуль, “Інформаційна технологія побудови інтелектуальних систем виявлення шахрайства при інсталюванні мобільних додатків”, *Інформаційні технології та комп'ютерна інженерія*, № 1, с. 4-16, 2019.

[12] T. Polhul, “Development of an intelligent system for detecting mobile app install fraud”, in *Proc. IRES 156th International Conference*, Bangkok, Thailand, 2019, pp. 25-29.

[13] А.А. Яровий, та Т.Д. Польгуль, “Комп'ютерна програма “Програмний модуль збору даних інформаційної технології виявлення шахрайства при інсталюванні програмних додатків”, *Свідоцтво про реєстрацію авторського права на твір № 76348*, К.: Міністерство економічного розвитку і торгівлі України, 2018.

[14] А.А. Яровий А. А., та Т.Д. Польгуль, “Комп'ютерна програма “Програмний модуль визначення схожості користувачів інформаційної

технології виявлення шахрайства при інсталюванні програмних додатків", *Свідоцтво про реєстрацію авторського права на твір № 76347*, К.: Міністерство економічного розвитку і торгівлі України, 2018.

[15] Т.Д. Польгуль, та А.А. Яровий, "Визначення шахрайських операцій при встановленні мобільних додатків з використанням інтелектуального аналізу даних", *Сучасні тенденції розвитку системного програмування. Тези доповідей*, Київ, 2016, с. 55-56. Режим доступу: http://ccs.nau.edu.ua/wp-content/uploads/2017/12/%D0%A1%D0%A2%D0%A0%D0%A1%D0%9F_2016_07.pdf

[16] А. Яровий, Т. Польгуль, та Л. Крилик, "Розробка методу виявлення шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних", *XIV Міжнародна конференція Контроль і управління в складних системах (КУСС-2018). Тези доповідей*, Вінниця, 2018, с. 35.

[17] А.А. Яровий, та Т.Д. Польгуль, "Подолання різнорідності вхідних даних при виявленні шахрайства при інсталюванні мобільних додатків з використанням інтелектуального аналізу даних", на *П'ятій міжнародній науково-технічній конференції студентів, магістрів, аспірантів «Інформатика, управління та штучний інтелект»*, Національний технічний університет «Харківський політехнічний інститут», Харків, 2018, с. 109.

[18] Т.Д. Польгуль, та А.А. Яровий, "Визначення шахрайських операцій при інсталяції мобільних додатків з використанням інтелектуального аналізу даних", на *XLVI науково-технічній конференції підрозділів ВНТУ*, Вінниця, 2017. Режим доступу: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/17200/2158.pdf?sequence=3

[19] Т.Д. Польгуль, "Моделювання процесу виявлення шахрайства при інсталюванні мобільних додатків", на *XLVIII науково-технічній конференції підрозділів ВНТУ*, Вінниця, 2019. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2019/paper/view/6863

[20] Т.Д. Польгуль, "Порівняльний аналіз Apache Spark та Apache Flink для роботи з Big Data", на *XLV науково-технічній конференції підрозділів ВНТУ*,

Вінниця, 2016. Режим доступу: https://ir.lib.vntu.edu.ua/handle/123456789/11619