**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Кваліфікаційна наукова
праця на правах рукопису

**САГАЙДАК ВІКТОР АНАТОЛІЙОВИЧ**

УДК 004.051

**ДИСЕРТАЦІЯ**

# МЕТОДИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВИЯВЛЕННЯ ШАХРАЙСТВА НА МОБІЛЬНІЙ МЕРЕЖІ ЗА ДОПОМОГОЮ КОМПЛЕКСНОГО ВИКОРИСТАННЯ CDR З РІЗНИХ ДЖЕРЕЛ

Спеціальність 123 «Комп’ютерна інженерія»
Галузь знань 12 «Інформаційні технології»
Подається на здобуття наукового ступеня
доктора філософії

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

Науковий керівник: СТОРЧАК Каміла Павлівна, доктор технічних наук, професор

Київ - 2024

# АНОТАЦІЯ

**Сагайдак В. А.** Методи підвищення ефективності виявлення шахрайства на мобільній мережі за допомогою комплексного використання CDR з різних джерел. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії в галузі знань 12 – Інформаційні технології, за спеціальністю 123 – Комп'ютерна інженерія. – Державний університет інформаційно-комунікаційних технологій. – Київ, 2024.

Дисертаційна робота присвячена актуальній науковій задачі підвищення ефективності функціонування мобільної мережі за рахунок зменшення обчислювального навантаження, що виникає під час шахрайської діяльності.

Тематика дисертаційного дослідження відповідає стандарту та фаховим компетентностям освітньо-наукової програми підготовки докторів філософії з комп'ютерної інженерії Державного університету інформаційно-комунікаційних технологій Міністерства освіти і науки України, а саме: фундаментальним науковим дослідженням теоретико-методологічних, науково-методичних та прикладних засад підвищення ефективності інноваційної та виробничої діяльності підприємства, а також вдосконаленню процесу забезпечення впровадження новітніх інформаційних технологій на об'єктах інформаційної діяльності.

*Актуальність дослідження* систем моніторингу шахрайства, а саме ефективність виявленні та ідентифікації зловмисницької діяльності, можна обґрунтувати наступним чином:

1. *Безпека мережі:* Шахрайство може коштувати втратою репутації, абонентів, фінансів та навіть відмову роботи самої мережі. Тому виявлення шахраїв дозволяє зменшити витрати на відновлення репутації, підвищити довіру користувачів мережі та своєчасно виявити спробу такої атаки.

2. *Економічний аспект:* У теперішній час технології розвиваються дуже стрімко. Дуже часто постає питання ефективності у використання того чи іншого програмно-апаратного комплексу.

3. *Комплексний вимір ключових показників виявлення:* Зазвичай швидкість виявлення залежить не тільки від модулів самої системи, скільки від її налаштування та типу даних, що надходять у систему. Останній аспект, з точки зору аналітики, є першочерговим показником.

Аналіз практичних підходів до систем виявлення шахрайства виявив, що ці системи використовують стандартизований формат даних NRTRDE, TAP3 та нестандартизований формат - безпосередній сбір з мережі. NRTRDE/TAP3 надають насичений об'єм необхідної інформації, але з затримкою не більше ніж 4 години або не більше ніж 30 днів. У такому випадку оператор, що надає інфокомунікаційні послуги, виявляє шахрайства з великою затримкою. Безпосередній збір з мережі має перевагу у майже відсутній затримці до 5 хвилин, але не має визначених недоліків.

Аналіз теоретичних підходів до оцінки ефективності виявлення шахрайства, показав що за недостатньо кількості досліджень та інформації стосовно процесу моніторингу та факторів впливу, більшість досліджень або поверхнево оцінюють весь процес, або ж дають оцінку ефективності на базі розроблених модулів виявлення конкретного виду шахрайства всередині самої систему моніторингу за допомогою машинного навчання, використовуючи заздалегідь підготовлений набір деталізованих записів, які не відповідають встановленим стандартам, форматам, процесам оператора інфокомунікаційної мережі. Тобто, іншими словами, перевірка ефективності та апробація методів проводилась у умовах, які не враховують час за який інформація була передана з однієї мережі до іншої, надійшла до системи, пройшла обробку та специфікацію типу цих деталізованих записів, що безпосередньо впливають на ефективність в цілому.

Тому, для підвищення ефективності було запропоновано та розроблено CDR (деталізованих записів) потік з IMS комутаторів базової мережі, який у комбінації з стандартизованими форматами дозволяє з меншою затримкою виявляти шахраїв. Для виміру ефективності були розроблені ключові показники та метод, що дозволяє виміряти час надходження та розпізнання.

Перший розділ дисертації присвячено аналізу складових систем, їх процесів та елементів у процесі роботи з даними. Проведено аналіз систем виявлення шахрайства на інфокомунікаційній мереж. Ці системи відіграють важливу роль у сучасних умовах, коли потреба у боротьбі з шахрайством стає все більш актуальною.

Було встановлено, що деякі системи моніторингу є досить інноваційними в плані реалізації за допомогою машинного навчання та використанні хмарного середовища, але підтримують лише одне джерело даних, доки інші системи використовують різноманітні типи сховищ реляційного та нереляційного типу, які використовуються в залежності від типу джерела даних. Тому, типи шахрайства потребують більш детального аналізу.

Для досягнення мети дослідження, а саме підвищення ефективності процесу виявлення шахрайської діяльності за рахунок комбінації потоку деталізованих записів з комутаторів разом з стандартизованими форматами, наприкінці розділу були сформовані наступні наукові завдання:

1. Дослідити методи порівняння даних з мережного зонду для виявлення негативного впливу на роботу мережних елементів.
2. Проаналізувати методи моніторингу даних віртуалізованого середовища з резервуванням.
3. Дослідити потік CDR даних з IMS комутаторів та розробити алгоритм взаємодії системи розрахунку з системою моніторингу.
4. Визначити складові архітектури системи аналітики великих даних у залежності від джерела інформації та розробити схему етапів виявлення шахрайства.
5. Розрахувати показники оцінки середньозваженого значення часу затримки для визначення ефективності розробленого інтерфейсу на тестовому середовищі, що імітує роботу інформаційної мережі.

У другому розділі проводиться аналіз впливу шахрайства на інфокомунікаційну мережу, основні види, технології, що використовуються для реалізації того чи іншого виду шахрайства та основні ознаки шахрайської атаки

або діяльності. Було проаналізовані складові NRTRDE та TAP3, передумови їх виникнення, типи сервісів для передачі та їх недоліки. Детальна увага була приділена безпосередньому збору даних з мережі за допомогою мережного зонду. Розглянуто такі технології як оптичні відгалужувачі та віддзеркалення трафіку з портів, наведено основні засади під час інтеграції з елементами мережі. Було досліджено аспекти інтеграції віртуалізованого середовища в інфокомунікаційну мережу та його експлуатацію, що здійснюють вплив на процес аналітики та виявлення шахрайства.

У третьому розділі дисертації розглянуто теоретичні аспекти та практичне застосування розробленого алгоритму за допомогою комплексного використання деталізованих записів. Були наведені елементи тестової мережі vEPC та схема взаємодії джерел даних з системою виявлення шахрайства на базі RDBMS Oracle. Була наведена загальна схема обробки трафіку та створено коефіцієнти визначення ефективності на основі часових проміжків з використанням середньозваженого значення. Деталізовані записи IMS платформи дозволили створити доповнений CDR, що може бути завантажений у БД для подальшого аналізу системою виявлення шахрайства. Основна увага приділяється програмній реалізації цього алгоритму, який базується на інтеграції bash кодування разом з інструментарієм ODI для трансформації формату полів та розрахунку наданих сервісів з наступним завантаження у БД Oracle. Розділ надає детальний опис кожного процесу та демонструє продуктивність розробленого інтерфейсу на основі методу визначення ефективності, який базується на використанні часових проміжків середньозваженого значення.

Були отримані наступні наукові результати:

1. Отримав подальшого розвитку метод моніторингу віртуалізованого середовища з резервуванням, який на відміну від існуючих дозволив виявити дублікацію даних, встановлення додаткового мережного зонду під час розширення мережі для удосконалення моделі підтримки її інфраструктури.

2. Розроблено алгоритм взаємодії IMS комутатора з системою виявлення шахрайства та розрахунку послуг, наукова новизна якого полягає у використанні

доступного bash кодування для форматування деталізованих записів, що базуються на застосуванні інструментів інтеграції даних, який дозволяє створити інтерфейс з наступним завантаженням інформації безпосередньо у базу даних системи моніторингу.

3. Вперше розроблено метод оцінки ефективності системи розпізнання шахрайства, що ґрунтується на статичному методі з використанням вагомого коефіцієнту, на основі комплексного використання деталізованих записів, який дозволив зменшити середньовагоме значення часу затримки даних у 3.7 разів для NRTRDE та у 14 разів для TAP3.

Дисертація виконувалась в Державному університеті інформаційно-комунікаційних технологій. Обраний напрям досліджень відповідає тематиці науково-дослідних робіт Державного університету інформаційно-комунікаційних технологій.

**Ключові слова:** Моніторинг, безпроводова мережа, інформаційна затримка, аналіз даних, статистичні моделі, текстова інформація, модель, система реального часу, машинне навчання, база даних, система виявлення вторгнень, контроль трафіку, інформаційна безпека, хмарні обчислення, статистичний аналіз.

# ABSTRACT

***Sahaidak V. A.*** Methods of improving the effectiveness of fraud detection on the mobile network by means of the integrated use of CDRs from various sources. – Qualifying scientific work as a manuscript.

Dissertation for obtaining the scientific degree of Doctor of Philosophy in the field of knowledge 12 - Information technologies, specialty 123 – Computer engineering. - State University of Information and Communication Technologies. - Kyiv, 2024.

The dissertation is devoted to the actual scientific task of increasing the efficiency of the mobile network by reducing the computing load that occurs during fraudulent activity.

The topic of the dissertation study corresponds to the temporary standard and professional competences of the educational and scientific program for training doctors of philosophy in computer engineering of the State University of Information and Communication Technologies of the Ministry of Education and Science of Ukraine, namely: fundamental scientific research of theoretical-methodological, scientific-methodological and applied principles of promotion the efficiency of the enterprise's innovative and production activities, as well as improving the process of ensuring the introduction of the latest information technologies at the objects of information activities.

The relevance of the study of fraud management systems (FMS), namely the effectiveness of detection and identification of malicious activity, can be justified as follows:

1. Network Security: Fraud can cost you reputation, subscribers, finances, and even network downtime. Therefore, detection of fraudsters allows to reduce the costs of reputation restoration, increase the trust of network users and detect an attempt of such an attack in a timely manner.

2. Economic aspect: Nowadays, technologies are developing very rapidly. Very often, the question of efficiency in the use of this or that software and hardware complex arises.

3. Comprehensive measurement of key detection indicators: Usually, the speed of detection depends not only on the modules of the system itself, but also on its configuration and the type of data entering the system. The last aspect, from the point of view of analytics, is a primary indicator.

Analysis of practical approaches to fraud detection systems found that these systems use the standardized data format NRTRDE, TAP3 and the non-standardized format - direct collection from the network. NRTRDE/TAP3 provide the saturated volume of information required, but with a delay of no more than 4 hours or no more than 30 days. In this case, the operator providing information communication services detects fraud with a long delay. Direct collection from the network has the advantage of an almost non-existent delay of up to 5 minutes, but has no identified disadvantages.

The analysis of theoretical approaches to the evaluation of the effectiveness of fraud detection showed that due to the insufficient number of studies and information regarding the monitoring process and influencing factors, the majority of studies either superficially evaluate the entire process, or give an evaluation of the effectiveness based on the developed modules for detecting a specific type of fraud within the monitoring system itself with the help of machine learning, using a pre-prepared set of detailed records that do not meet the established standards, formats, processes of the information communication network operator. That is, in other words, the verification of efficiency and approbation of methods was carried out in conditions that do not take into account the time during which information was transferred from one network to another, arrived at the system, underwent processing and specification of the type of these detailed records, which directly affect the efficiency as a whole.

Therefore, to improve efficiency, a CDR (call detail record) flow from the IMS of the core network switches was proposed and developed, which, in combination with standardized formats, allows detecting fraudsters with less delay. Key indicators and a

method to measure arrival and recognition time were developed to measure performance.

The first chapter of the dissertation is devoted to the analysis of component systems, their processes and elements in the process of working with data. The analysis of fraud detection systems on information communication networks was carried out. These systems play an important role in today's environment, when the need to fight fraud is becoming more and more urgent.

It has been found that some monitoring systems are quite innovative in terms of implementation through machine learning and use of the cloud environment, but support only one data source, while other systems use a variety of relational and non-relational storage types, which are used depending on the type of data source. Therefore, the types of fraud require a more detailed analysis.

To achieve the goal of the research, namely to increase the effectiveness of the process of detecting fraudulent activity due to the combination of the flow of detailed records from switches together with standardized formats, the following scientific tasks were formed at the end of the chapter:

1. Investigate methods of comparing data from a network probe to identify a negative impact on the operation of network elements.
2. Analyze data monitoring methods of a virtualized environment with redundancy.
3. Investigate the flow of CDR data from IMS switches and develop an algorithm for the interaction of the calculation system with the monitoring system.
4. Determine the components of the architecture of the big data analytics system depending on the source of information and develop a scheme of fraud detection stages.
5. Calculate the evaluation indicators of the weighted average value of the delay time to determine the effectiveness of the developed interface on a test environment that simulates the operation of an information network.

The second section analyzes the impact of fraud on the information and communication network, the main types, technologies used to implement one or another

type of fraud, and the main signs of a fraudulent attack or activity. The components of NRTRDE and TAP3, the prerequisites for their occurrence, the types of services for transmission and their shortcomings were analyzed. Detailed attention was paid to the direct collection of data from the network using a network probe. Such technologies as optical splitters and mirroring of traffic from ports are considered, the basic principles during integration with network elements are given. Aspects of the integration of the virtualized environment into the information communication network and its operation, which influence the process of analytics and fraud detection, were investigated.

In the third chapter of the dissertation, the theoretical aspects and practical application of the developed algorithm are considered with the help of complex use of detailed records. The elements of the vEPC test network and the scheme of interaction of data sources with the fraud detection system based on RDBMS Oracle were given. A general scheme of traffic processing was given and performance coefficients were created based on time intervals using a weighted average value. The detailed records of the IMS platform made it possible to create a supplemented CDR, which can be loaded into the database for further analysis by the fraud detection system. The main attention is paid to the software implementation of this algorithm, which is based on the integration of bash coding together with the ODI toolkit for the transformation of the field format and the calculation of the provided services, followed by uploading to the Oracle database. The section provides a detailed description of each process and demonstrates the performance of the developed interface based on the time-weighted average performance method.

The following scientific results were obtained:

1. The method of monitoring a virtualized environment with redundancy received further development, which, unlike the existing ones, made it possible to detect data duplication, install an additional network probe during network expansion to improve the model of supporting its infrastructure.

2. An algorithm for the interaction of the IMS switch with the fraud detection and service calculation system has been developed, the scientific novelty of which is the use of available bash coding for formatting detailed records based on the application of

data integration tools, which allows creating an interface with subsequent uploading of information directly into the monitoring system database.

3. For the first time, a method for evaluating the effectiveness of a fraud detection system based on a static weighting method was developed, based on the comprehensive use of detailed records, which allowed to reduce the weighted average data delay time by 3.7 times for NRTRDE and 14 times for TAP3.

The dissertation was completed at the State University of Information and Communication Technologies. The chosen direction of research corresponds to the topic of research works of the State University of Information and Communication Technologies.

**Key words:** Monitoring, wireless network, information latency, data analysis, statistical models, text information, model, real-time system, machine learning, database, intrusion detection system, traffic control, information security, cloud computing, statistical analysis.

# СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

*Наукові праці, у яких опубліковані основні наукові результати дисертації:*

1. Алтинніков Д. Є., Шевченко О. О., Бердник І. І., Зуб О. В., Сагайдак В. А., «Використання Java--анотацій як інструменту надання API», *Зв'язок*, № 4(152), с. 56–59, 2021.

2. Сагайдак В. А., Сеньков О. В., «Huawei Genex Discovery – інструмент виявлення великих даних для аналізу безпроводової мережі», *Зв'язок*, № 4(158), с. 34–41, 2022.

3. Сагайдак В. А., Лисенко М. М., Сеньков О. В., «Шахрайство у сфері телекомунікацій та його вплив на бізнес операторів зв'язку», *Зв'язок*, № 6(160), с. 17–20, 2022.

4. Сагайдак В. А., «Огляд систем розпізнання шахрайства та розробка коефіцієнтів для визначення їх ефективності», *Кібербезпека: освіта, наука, техніка*, № 3 (23), с. 274-283, 2024.

5. Сачук О. В., Сагайдак В. А., «Розроблення методики транскрибації на основі нейронних мереж», *Зв'язок*, № 2(168), с. 23-26, 2024.

6. IX Науково-технічна конференція студентів та молодих вчених факультету Інформаційних технологій «Сучасні інфокомунікаційні технології»; Система для аналізу та моніторингу радіопокриття базових станцій; 11 грудня 2020, Державний університету телекомунікацій, м. Київ.

7. IV Всеукраїнська науково-практична конференція «Telecommunication: problems and innovation»; SMS fraud realization and recognition methods; 30 травня 2023 р., Державний університету телекомунікацій, м. Київ.

8. V Всеукраїнська науково-практична конференція «Telecommunication: problems and innovation»; TAP3 and NRTRDE CDR transfer formats; 20 грудня 2023р., Державний університет інформаційно-комунікаційних технологій, м. Київ.

9. XIII міжнародна науково технічна конференція The 13th International Scientific Conference «ITSEC»; Rhino IMS CDR APV fields for network and subscriber

identification; 9-11 травня 2024 р., Львівський національний університет імені Івана Франка, м. Львів.

# ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Big Data – Великі дані

FMS – (Fraud Management System) система виявлення шахрайства

IMS – (IP Multimedia subsystem) мультимедійна система на основі IP

EPC – (Evolved Packet Core) базова мережа LTE

CDR – (Call Detailed Record) деталізований запис дзвінка

IEEE- Інститут інженерів по радіотехніці і електроніці

RDBMS - (Relational Database Management System) система керування реляційною базою даних

SQL – (Structured Query Language) мова структурованих запитів

ETL – (Extract Transform Load) витягнути, трансформувати та завантажити

LTE – (Long Term Evolution) мережа радіо доступу 4G

VoLTE – (Voice over LTE) голосові сервіси LTE

NRTRDE – (Near Real Time Roaming Data Exchange) обмін даними роумінгу у режимі наближеному до реального часу

TAP3 – (Transferred Account Procedure 3) обмін даними для виплати

NFV – (Network Functions Virtualization) віртуалізація мережевих функцій

AVP – (Attribute Value Pair) атрибут, що об’єднує визначення та його значення

VM – (Virtual Machine) віртуальна машина

ODI – (Oracle Data Integrator) інструмент для взаємодії інформації та базою даних

# ВСТУП

Дисертаційне дослідження присвячено аналізу систем моніторингу шахрайства на мобільній мережі та їх ефективності у виявленні та ідентифікації зловмисницької діяльності за допомогою різних джерел даних. Ці системи відіграють важливу роль у сучасних умовах, коли потреба у ефективному виявленню шахрайства стає все більш актуальною.

*Актуальність дослідження* систем моніторингу шахрайства, а саме ефективність виявленні та ідентифікації зловмисницької діяльності, можна обґрунтувати наступним чином:

1. *Безпека мережі:* Шахрайство може коштувати втратою репутації, абонентів, фінансів та навіть відмову роботи самої мережі. Тому виявлення шахраїв дозволяє зменшити витрати на відновлення репутації, підвищити довіру користувачів мережі та своєчасно виявити спробу такої атаки.

2. *Економічний аспект:* У теперішній час технології розвиваються дуже стрімко. Дуже часто постає питання ефективності у використання того чи іншого програмно-апаратного комплексу.

3. *Комплексний вимір ключових показників виявлення:* Зазвичай швидкість виявлення залежить не тільки від модулів самої системи, скільки від її налаштування та типу даних, що надходять у систему. Останній аспект, з точки зору аналітики, є першочерговим показником.

*Аналіз практичних підходів* до систем виявлення шахрайства виявив, що ці системи використовують стандартизований формат даних NRTRDE, TAP3 та нестандартизований формат - безпосередній сбір з мережі. NRTRDE/TAP3 надають насичений об’єм необхідної інформації, але з затримкою не більше ніж 4 години або не більше ніж 30 днів. У такому випадку оператор, що надає інфокомунікаційні послуги, виявляє шахрайства з великою затримкою. Безпосередній збір з мережі має перевагу у майже відсутній затримці до 5 хвилин, але не має визначених недоліків.

*Аналіз теоретичних підходів* до оцінки ефективності виявлення шахрайства, показав що за недостатньо кількості досліджень та інформації стосовно процесу моніторингу та факторів впливу, більшість досліджень або поверхнево оцінюють весь процес, або ж дають оцінку ефективності на базі розроблених модулів виявлення конкретного виду шахрайства всередині самої систему моніторингу за допомогою машинного навчання, використовуючи заздалегідь підготовлений набір деталізованих записів, які не відповідають встановленим стандартам, форматам, процесам оператора інфокомунікаційної мережі. Тобто, іншими словами, перевірка ефективності та апробація методів проводилась у умовах, які не враховують час за який інформація була передана з однієї мережі до іншої, надійшла до системи, пройшла обробку та специфікацію типу цих деталізованих записів, що безпосередньо впливають на ефективність в цілому.

Тому, для підвищення ефективності було запропоновано та розроблено CDR (деталізованих записів) потік з IMS комутаторів базової мережі, який у комбінації з стандартизованими форматами дозволяє з меншою затримкою виявляти шахраїв. Для виміру ефективності були розроблені ключові показники та метод, що дозволяє виміряти час надходження та розпізнання.

*Об’єкт дослідження* – процес виявлення шахрайства за допомогою системи моніторингу.

*Предмет дослідження* – методи та алгоритми виявлення шахрайства у системі моніторингу.

*Мета дослідження* полягає у підвищенні ефективності процесу виявлення шахрайської діяльності за рахунок комбінації потоку CDR даних з IMS комутаторів разом з стандартизованими форматами.

Були сформовані наступні *наукові завдання:*

1. Дослідити методи порівняння даних з мережного зонду для виявлення негативного впливу на роботу мережних елементів.
2. Проаналізувати методи моніторингу даних віртуалізованого середовища з резервуванням.

3. Дослідити потік CDR даних з IMS комутаторів та розробити алгоритм взаємодії системи розрахунку з системою моніторингу.

4. Визначити складові архітектури системи аналітики великих даних у залежності від джерела інформації та розробити схему етапів виявлення шахрайства.

5. Розрахувати показники оцінки середньозваженого значення часу затримки для визначення ефективності розробленого інтерфейсу на тестовому середовищі, що імітує роботу інформаційної мережі.

*Методи дослідження* – аналіз, моделювання, статистичний, структурний, кореляційний, графічний, експериментальні дослідження.

*Наукова новизна:* Вперше розроблена методика оцінки ефективності системи розпізнання шахрайства на основі комплексного використання CDR з різних джерел даних, дозволяє дати оцінку процесу моніторингу з урахуванням типу даних та часу їх надходження у систему.

Для досягнення поставленої мети дисертаційної роботи були вирішені такі завдання:

1. Удосконалено модель середовища обробки даних за допомогою записів з комутаторів базової мережі, що дозволяє пришвидшити обробку інформації.

2. Набув подальшого розвитку метод підвищення ефективності розпізнання шахрайства за допомогою даних CDR з комутаторів базової мережі, що дозволяє підвищити ефективності функціонування мобільної мережі за рахунок зменшення обчислювального навантаження, що виникає під час шахрайської діяльності.

3. Вперше розроблено методику оцінки ефективності системи розпізнання шахрайства, що ґрунтується на статичному методі з використанням вагомого коефіцієнту, на основі комплексного використання деталізованих записів.

*Особистий внесок здобувача:* Основні положення та результати дисертаційної роботи отримані автором самостійно. Автор виконав усі теоретичні та практичні дослідження, що становлять основу дисертаційної роботи.

В опублікованих роботах у співавторстві, згідно списку опублікованих праць за темою дисертації (с.12-13), здобувачу належать такі результати: [1,2,4,6] – проаналізовано та виявлено основні етапи обробки в системах Big data на основі яких була створена методологія на основі ключових показники виявлення ефективності; [3,7] – наведено найбільш розповсюджені методи шахрайства та їх вплив на оператора інфокомунікаційної мережі; [5,8] – дослідження основних методів передачі та збору з інформаційної мережі, були розглянуті їх сценарії застосування, недоліки та переваги; [9] – дослідження складових CDR IMS системи, що використовуються для ідентифікації мережі та абонентів;

*Апробація матеріалів дисертації:*

Основні результати дисертаційної роботи доповідалися і обговорювалися на 3 науково-технічних та науково-практичних конференціях:

1. IX Науково-технічна конференція студентів та молодих вчених факультету Інформаційних технологій «Сучасні інфокомунікаційні технології», 11 грудня 2020, Державний університету телекомунікацій, м. Київ.

2. IV Всеукраїнська науково-практична конференція «Telecommunication: problems and innovation», 30 травня 2023 р., Державний університету телекомунікацій, м. Київ.

3. V Всеукраїнська науково-практична конференція «Telecommunication: problems and innovation», 20 грудня 2023р., Державний університет інформаційно-комунікаційних технологій, м. Київ.

4. XIII міжнародна науково технічна конференція The 13th International Scientific Conference «ITSEC», 9-11 травня 2024 р., Львівський національний університет імені Івана Франка, м. Львів.

*Структура та обсяг дисертації :* дисертація складається з анотації, змісту, переліку умовних скорочень вступу, трьох розділів, загальних висновків, списку використаних джерел та додатків і має 160 сторінок основного тексту, 74 рисунків

та таблиць, 15 сторінок додатків. Список використаних джерел містить 107 найменувань і займає 11 сторінок. Загальний обсяг дисертаційної роботи – 201 сторінка.

*Практична цінність:* результати дослідження можуть бути використані при інтеграції та оцінки системи аналітики для виявлення шахрайства.

*Застосування результатів роботи:* Результати наукових досліджень були використані та впроваджені у проекті на підприємстві ТОВ "ІНФОПУЛЬС УКРАЇНА".

# 1 ОГЛЯД ОСНОВНИХ СКЛАДОВИХ ETL СИСТЕМИ ВИЯВЛЕННЯ ШАХРАЙСТВА ТА BIG DATA, ЇХ РЕАЛІЗАЦІЯ В ЗАЛЕЖНОСТІ ВІД ДЖЕРЕЛ ІНФОРМАЦІЇ У СФЕРІ ЗАСТОСУВАННЯ

## 1.1 Огляд систем моніторингу

### 1.1.1 Принцип роботи системи на базі AWS сервісів

AWS Fraud Detection [4,45] забезпечує обробку даних NRTRDE за допомогою машинного навчання. Дане рішення працює наступним чином (рис. 1.1):

1) Під час навчання штучного інтелекту (ШІ) інфокомунікаційні дані передаються пачками або потоково, використовуючи Amazon Kinesis, у відро Amazon Simple Storage Service (Amazon S3) за допомогою AWS Glue Data Catalog, яке виконує функції індексації інформації.
2) Дані оброблюються за допомогою Amazon SageMaker Data Wrangler і перетворюються на функції. Дані можна отримати безпосередньо з Amazon S3 або за допомогою запитів Amazon Athena. Функції зберігаються в Amazon SageMaker Feature Store.
3) Amazon SageMaker навчає спеціальну (класифікаційну) модель для виявлення шахрайства. Для переконання того, що модель упорядкована та ефективна для використання в реальному середовища, її тестують та перевіряють.
4) Навчену модель зберігають в Amazon SageMaker Model Registry для відстеження та керуванням нею з часом.
5) Amazon SageMaker Model Monitor використовується для відстеження якості моделі з плином часу, включно з даними і якість моделі, а також дрейф зміщення.
6) Після проходження всіх тестів на точність і продуктивність, модель встановлюється за допомогою кінцевих точок Amazon SageMaker для підтримки результатів майже в реальному часі. Кінцеві точки Amazon

SageMaker допомагають у керуванні масштабованості, ефективності роботи та надійністю.

7) Під час визначення результатів, дані від оператора інфокомунікаційних послуг та партнерів передаються потоково за допомогою Amazon Kinesis до функції AWS Lambda. Amazon API Gateway надає функціонал для керування доступом до кінцевих точок моделі. Результати виявлення шахрайства, створені за допомогою моделі, може бути використані оператором інформаційної мережі.

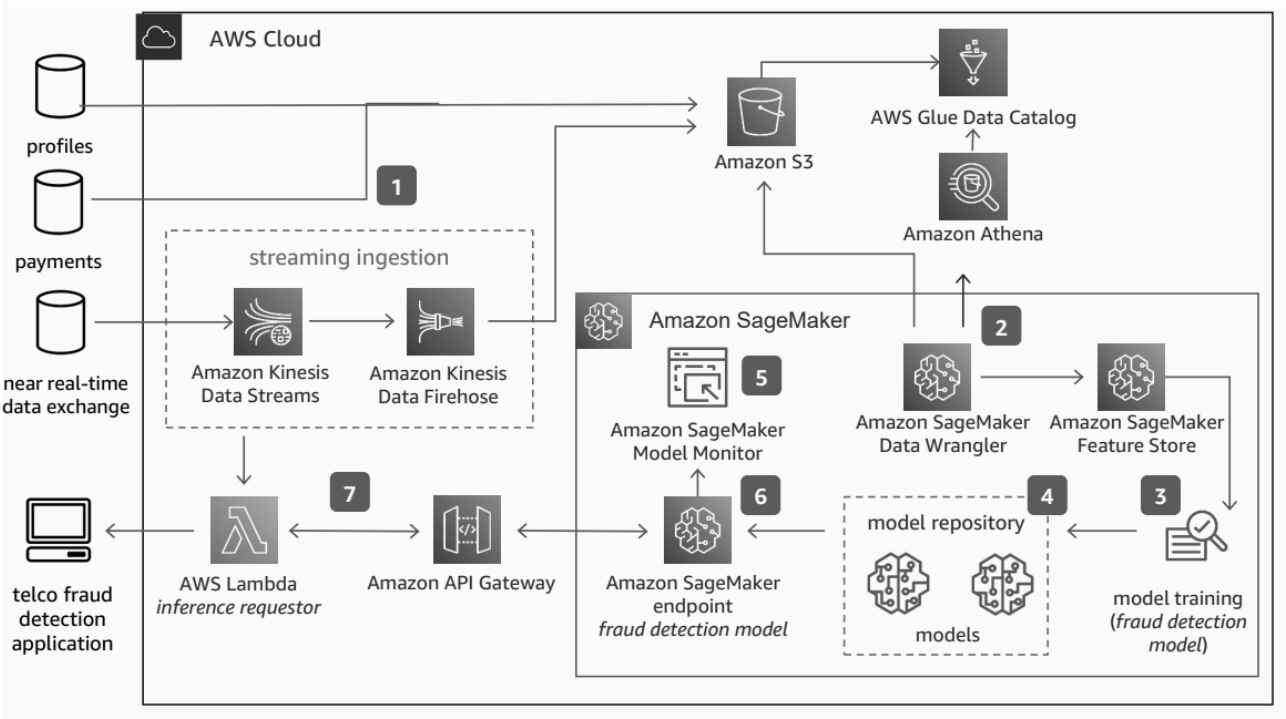



Рисунок 1.1 – Схема роботи AWS Fraud Detection

### 1.1.2 Принцип роботи систем Big Data на базі Hadoop

Інформаційні системи, які безпосередньо дані з мережі у режимі реального часу, зазвичай складаються з двох частин. Перша частин - підсистеми пасивного збору та насичення інформації. Після того як підсистема опрацює дані, друга частина, яка називається аналітична платформа, аналізує отримані дані та видає результат кінцевому споживачу.

Прикладом таких систем Big data можна навести кооперацією компаній Gigamon та Argyle Data, FraudView від Cvidya Amdocs, система розпізнання шахрайства від Subex, SmartCare та GENEX Discovery від компанії Huawei.

Комплекс розроблений Gigamon та Argyle Data [4,44] складається з Gigamon fabric для збору, фільтрування, доповнення інформації та системи розпізнання шахрайства Argyle Data Real-Time Fraud Analytics Hadoop Application, що побудована на технології Hadoop для зберігання зібраних даних та результати аналізу додатку (рис. 1.2).

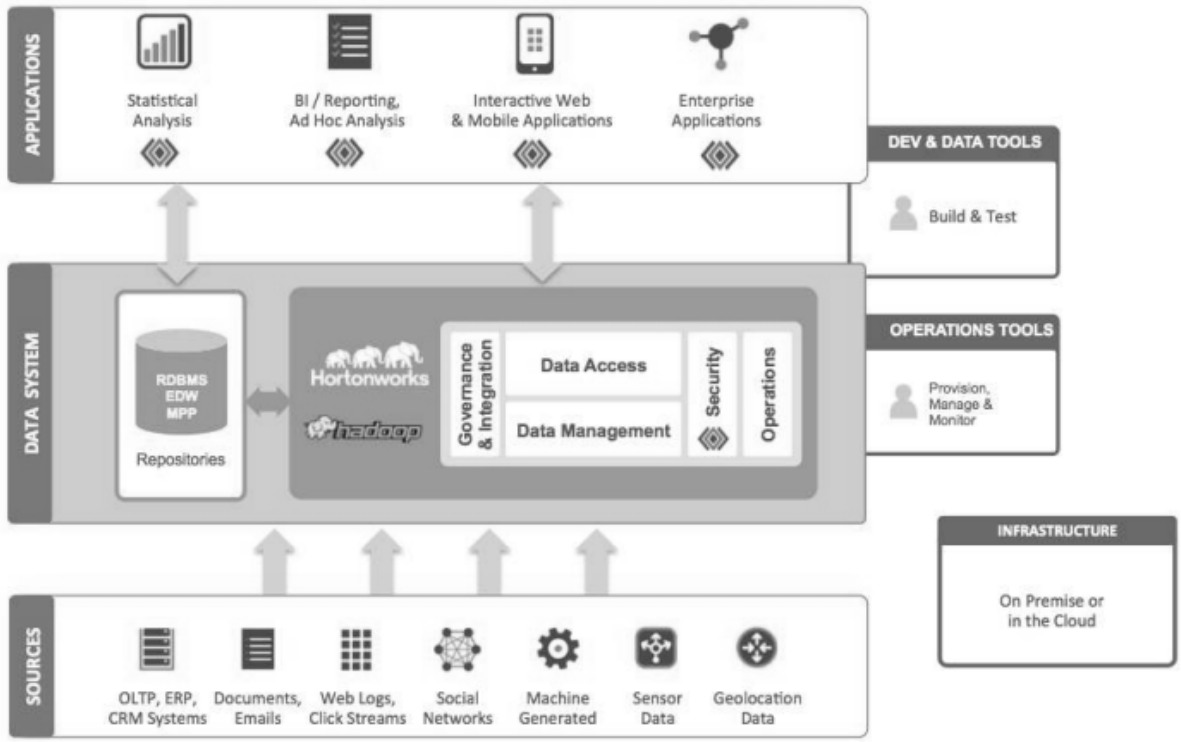



Рисунок 1.2 – Структура Argyle Data Real-Time Fraud Analytics Hadoop Application

Працює система на базі Hadoop наступним чином:

1) Спочатку інформація потрапляє на data access.
2) Потім за допомогою RDBMS (Relational database management system) дані записуються та зберігаються в Hadoop.
3) Наступним кроком додатки роблять запити на RDBMS, оброблюють та записують результат назад у Hadoop. Такими додатками можуть бути

статичний аналіз, звіти, BI (Business intelligence), аналітика по запиту, додатки на рівні підприємства, інтерактивні Web та мобільні додатки.

FraudView від Cvidya Amdocs (рис.1.3) збирає інформацію з різних точок як OSS/BSS, CRM, білінгових платформ, HLR, CDR з комутаторів, Probe (SS7, VoIP, IP) та обробляє її різними механізмами виявлення [4,48].

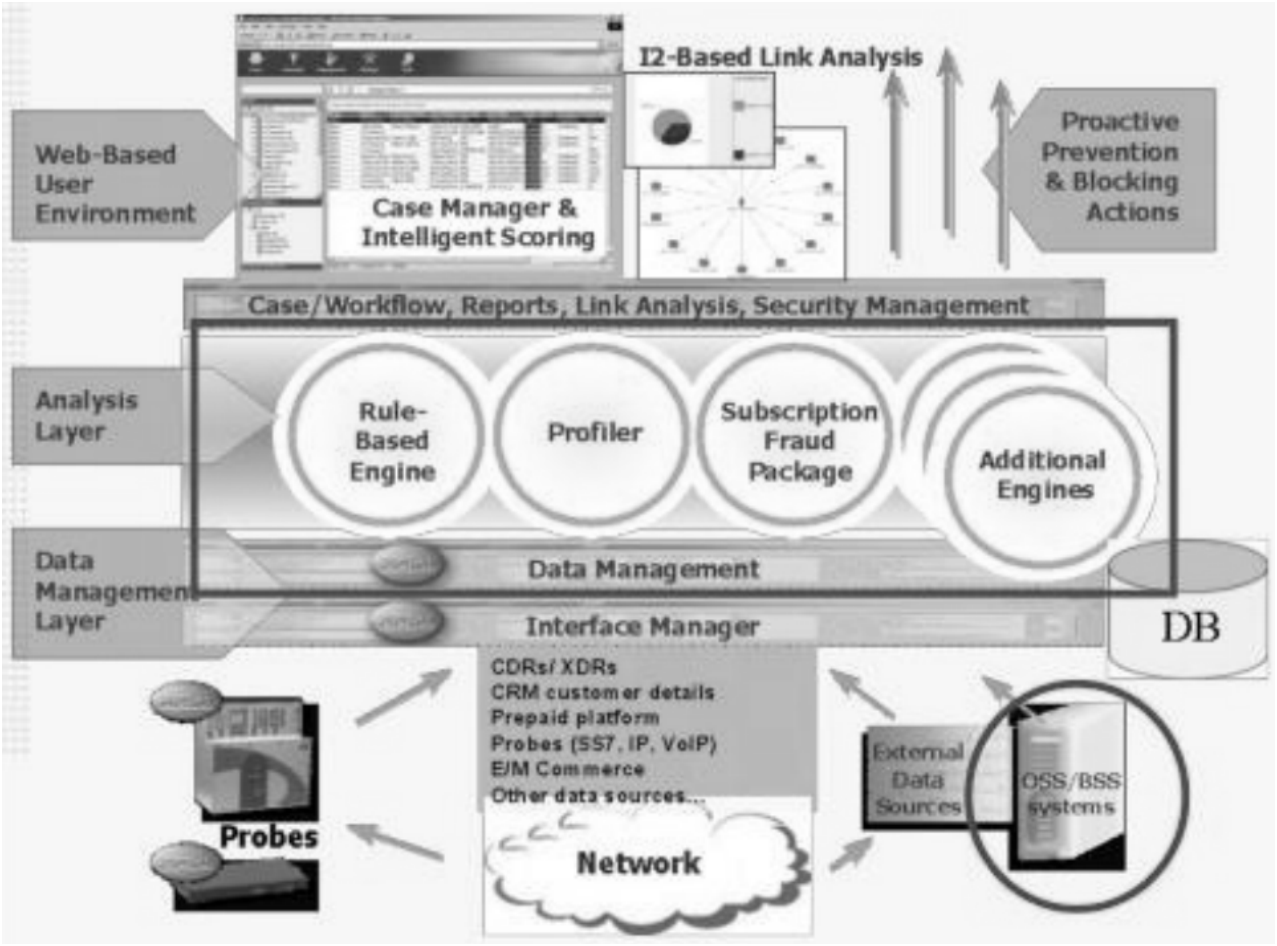



Рисунок 1.3 – Структура FraudView

Система працює наступним чином (рис.1.4):

1) FMS збирає інфомацію з різних джерел на рівні керування інтерфейсом (interface manager) та на рівні керування даними (data management) завантажує їх в БД.
2) Інформація з БД надається двигунам (engines), кожен з яких оброблює інформацію різними способами, а саме за допомогою правил, профілювання, тощо;

3) Після того, як один з двигунів закінчив обробку, створюється оповіщення про шахрайство та додається у справи про шахрайство.
4) Якщо ще якийсь інший двигун виявив шахрайство щодо суб'єкта, який підозрюється у шахрайстві та був помічений системою до того часу, то оповіщення про шахрайство додається до існуючої справи.
5) На рівні аналітики шахрайства (case analysis stage) система виявлення шахрайства надає аналітику CDR, клієнтської інформації (дані з білінгу), аналітику поведінки, аналітику звязку з іншими субєктами шахрайства, аналітику ознаки шахрайства, схеми шахрайства та аналітику на базі історичної поведінки (минулі сеанси зв'язку та аналітика оплати)

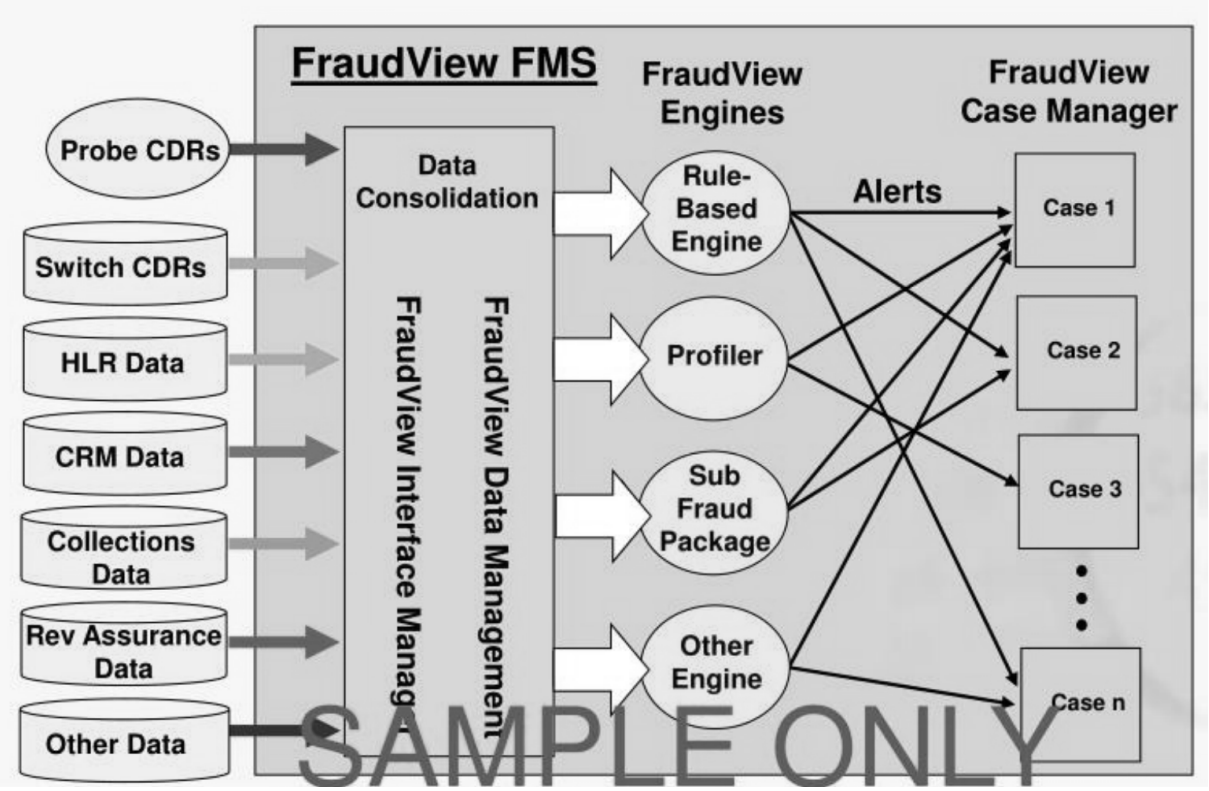



Рисунок 1.4 – FraudView структурна схема обробки інформації

Система розпізнання шахрайства (Fraud Management System - FMS) від компанії Subex (рис. 1.5) за допомогою гнучких та масштабуємих інструментів підготовки даних ETL (Extract, Transform, Load) в режимі реального часу з різних джерел даних, трансофрмує та доовнює їх [4,46]. Наступним кроком інформація

завантажується в БД. Ці дані оброблюються правилами, що можуть бути налаштовані на системі. Користувач має можливість корегувати правила, а саме правила з порогами відроботки, географічні правила, правила машинного навчання, правила з шаблонною поведінкою, правила з використанням гарячого списку, правила виявлення спаму, правила виявлення вторгнень, правила з розумним виявленням шаблоної поведінки. В кінці інформація, що була визначена системою як шахрайство, зображується на графіках та на їх основі створюються справи щодо зловмисницкої діяльності.

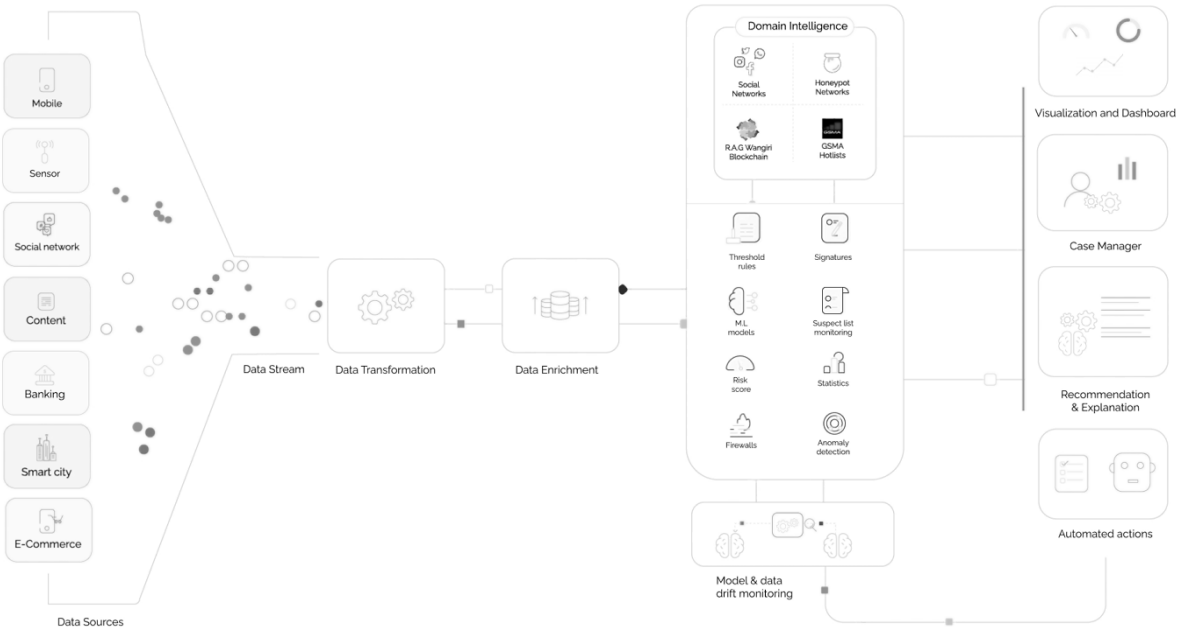



Рисунок 1.5 –Структурна обробки інформації Subex FMS

Huawei SmartCare та GENEX Discovery складаються з системи аналітики даних та підсистем Huawei PS Probe, Huawei CS Probe, Huawei CHR/MR, DGW, мережевого зонду від іншого виробника [4,10,11,12,13].

Huawei GENEX Discovery - це система для моніторингу статусу та аналізу радіо мережі, а саме GSM, UMTS та LTE. Кожна функція даного програмно-апаратного продукту може надавати статистичні дані стільника або базової станції з точки зору точки мережі, мережі сусідніх стільників, розташування, обладнання виробника або навіть терміналу користувача в режимі реального часу. Можливості оцінки на рівні мережі проводять оцінку за кількома параметрами такими як

покриття, продуктивність, трафік та оцінює стан мережі в поєднанні з результатами географічного спостереження, що допомагає користувачам швидко зрозуміти стан мережі та тенденції. Функція оцінки на рівні області надає стан працездатність кожної оцінюваної ділянки в мережі. Що допомагає користувачам швидко визначити проблемні регіони. Також, функція дозволяє користувачам встановлювати деякі області як пріоритетні, де значення лічильника якість значною мірою впливає на оцінку працездатності інших ділянок і всієї мережі. Основні елементи системи можна побачити на рисунку 1.6.

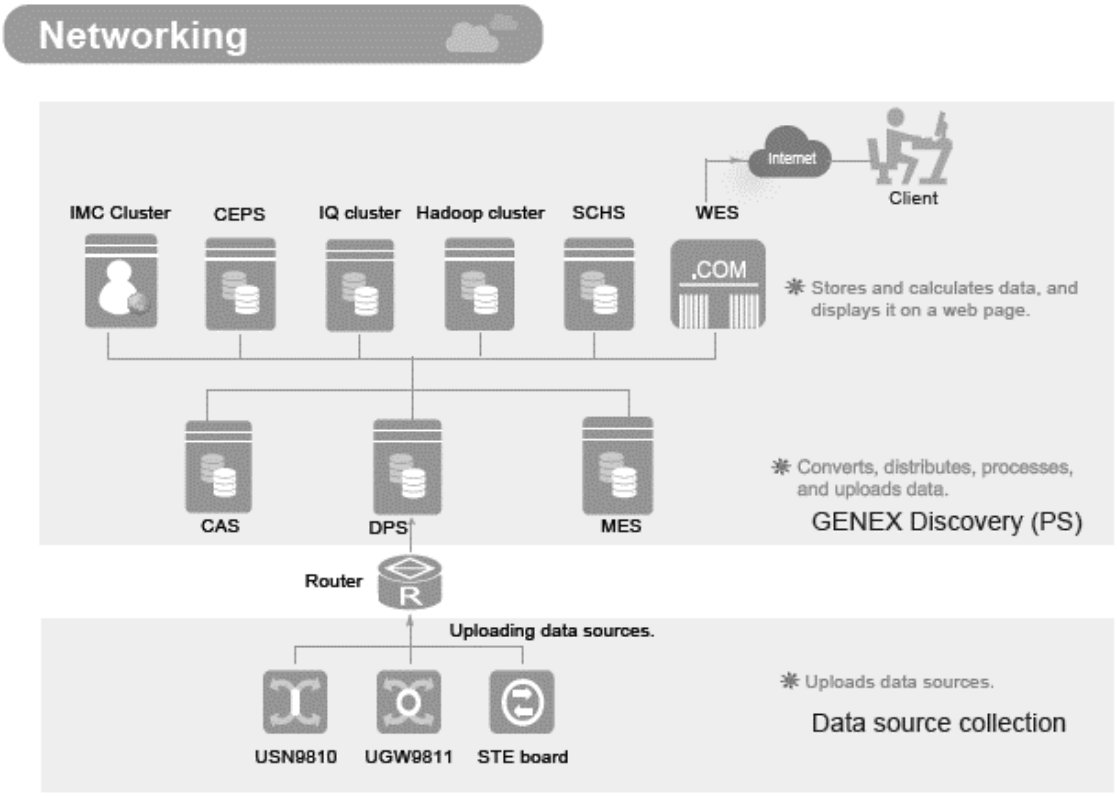



Рисунок 1.6 – мережеве розташування елементів GENEX Discovery

Кожен елемент GENEX Discovery виконує наступні функції:

- Веб сервер (WES) надає аналітичний звіт спостереження статусу сервесів в режимі реального часу, аналітику історичних даних та багатовимірну поглиблення в дані. Також надає функції такі як керування журналами подій, помилок, тощо та керування аваріями.

- Сервер планування (SCHS) надає функції керування задачами та їх розкладом роботи, керування метаданими
- Hadoop сервер (HDPS) зберігає усі CDR та спрощені CDR та розраховує добові SDR
- Sybase IQ кластер завантажує, обробляє та зберігає service detail records (SDRs)
- Сервер комплексної обробки подій (CEPS) ассоціює комплекс подій та спрощує CDR
- Кластер внутрішньопам'ятної обробки (IMC) розраховує базові ключові показники ефективності, SDR та має гнучку підтримку різних додатків, що знаходятся на більш високих рівнях
- Сервер аналітики корреляції (CAS) встановлює взаємозв'язок та консолідує CDR у формати, які можуть бути роспізнані модулями, що знаходятся на рівень вище.
- Сервер медіації (MES) конвертує вбудовані джерела даних (CHRs, UFDRs, та xDRs) у формати, які система може роспізнати.
- Сервер розподілення обробки (DPS) функціонує як модуль розсилки який отримує історичні дані дзвінка та розприділяє їх до різних серверів.

Розташування елементів GENEX Discovery може бути скорегована в залежності від пропускної здатності мережі.

Huawei SmartCare складається з двох підсистем – SEQ Analyst та NetProbe. SEQ Analyst (Service & Experience Quality Analyst) дозволяє ефективно керувати якістю обслуговування та продуктивністю мережі, швидкою обробкою скарг клієнтів, підтримку компанії маркетингу. Ця підсистема може розпізнавати E2E з мобільної мережі, транспортної мережі з послугами та додатками, які надають ці послуги.

Додатки для аналізу трафіка у SEQ Analyst можна розділити на три види:

- Керування якістю наданням послуг (Service Quality Management SQM) дозволяє ідентифікувати такі сервіси як Voice, SMS, HTTP, Email, Video,

FTP, VoLTE. SQM дозволяє порахувати ключові показники продуктивності та, на основі цих індикаторів, вивести статистику по регіонам, користувацьким пристроям, вебсайтам, мережевим пристроям базової мережі, а також вивести статистику по причинам відмов надання послуг. Ключові індикатори продуктивності можливо редагувати по принципу правило плюс механізм підрахунку. Розробка цих індикаторів стає можливою у спеціальному візуальному інструменті, що постачається окремо.

- Керування клієнтським досвідом (Customer Experience Management CEM) надає статистику якості послуг у режимі реального часу. Ця статистика надається у вигляді виділеного звіту по кожному користувачу мережі, по якості обслуговування. Коли у абонента виникають проблеми, система може надіслати оповіщення протягом однієї хвилини та показати на карті місце, де виникає різниця у якості обслуговування. Ділові записи та повідомлення сигналізації під час встановлення сеансу зв'язку (xDR, CDR) операторам інфокомунікацій вирішити проблеми та забезпечити обслуговування, перш ніж користувачі поскаржаться.
- Керування продуктивністю мережі (Network Performance Management NMP) дозволяє планування розвитку та оптимізацію мережі, розрахувати середній дохід на абонента, подивитися ємність мережі на послугу, спостерігати стан елементів транспортної мережі на рівні елементів , так і на рівні сигналізації.

SEQ Analyst працює по наступній схемі (рис. 1.7):

Спочатку трафік, який пройшов обробку в пробі або з інших елементів, надходить на систему попередньої обробки xDR (xDetail record). Після цього обробленні дані відправляються на модуль комплексної обробки даних канальної чи пакетної комутації частин транспортної мережі. Після цього результат записується у таблиці, які зберігаються у Data Warehouse. Їх можливо подивитися через спеціальний користувацький веб інтерфейс. Також, є можливість передачі

цих даних до зовнішніх систем, відмінних від SEQ Analyst, через підсистему обміну даними (Data Sharing Subsystem).

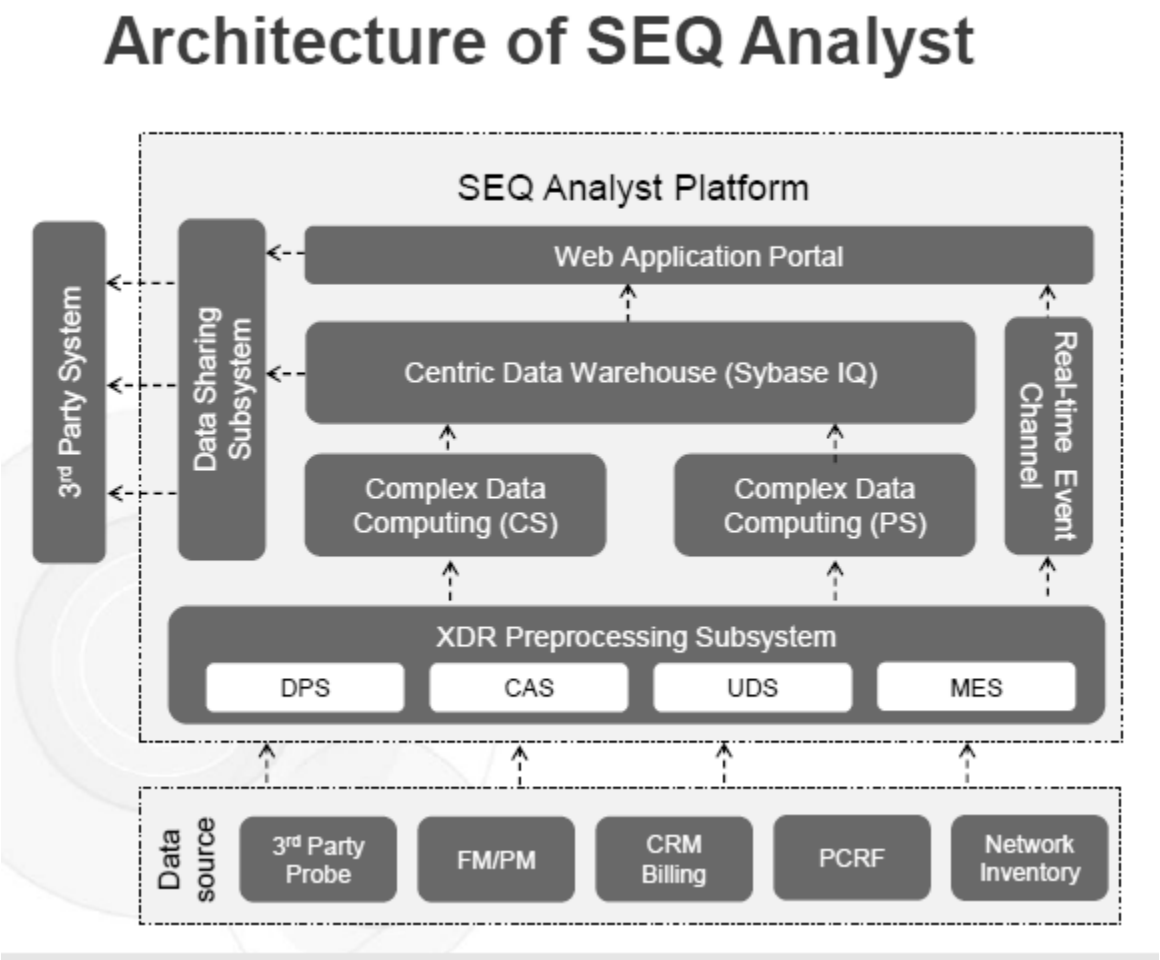



Рисунок 1.7 – Архітектура SEQ Analyst

SEQ Analyst має схожі елементи як у GENEX Discovery. Основна відмінність у архітектурі полягає, що Sybase IQ кластер та Greenplum використовується як сервер для зберігання усіх CDR, спрощених CDR, розрахунку добових SDR та завантаження, обробки та зберігання SDR. Також є розподілення у комплексній обробці між даними транспортної мережі канальної та пакетної комутації. З'являється підсистема обміну даними (DSS) з зовнішніми системами, відмінними від SEQ Analyst.

## 1.2 Огляд складових розглянутих систем

### 1.2.1 Складові системи на базі AWS сервісів

Система складається з наступних елементів:

**Amazon Kinesis** – аналізує та оброблює потокові дані будь-якого масштабу як повністю контролюємо послуга. Цей сервіс може використовуватися (рис. 1.8) для створення додатків, що працюють у режимі реального часу такі як моніторинг додатків, розпізнання зловмисницької діяльності, тощо. За допомогою Kinesis користувач може збирати інформацію у реальному часі, таку як відео, аудіо, журнали роботи додатків, телеметричні дані Інтернету речей, кліки вебсайтів для машинного навчання, аналітики та інших додатків [54,67].

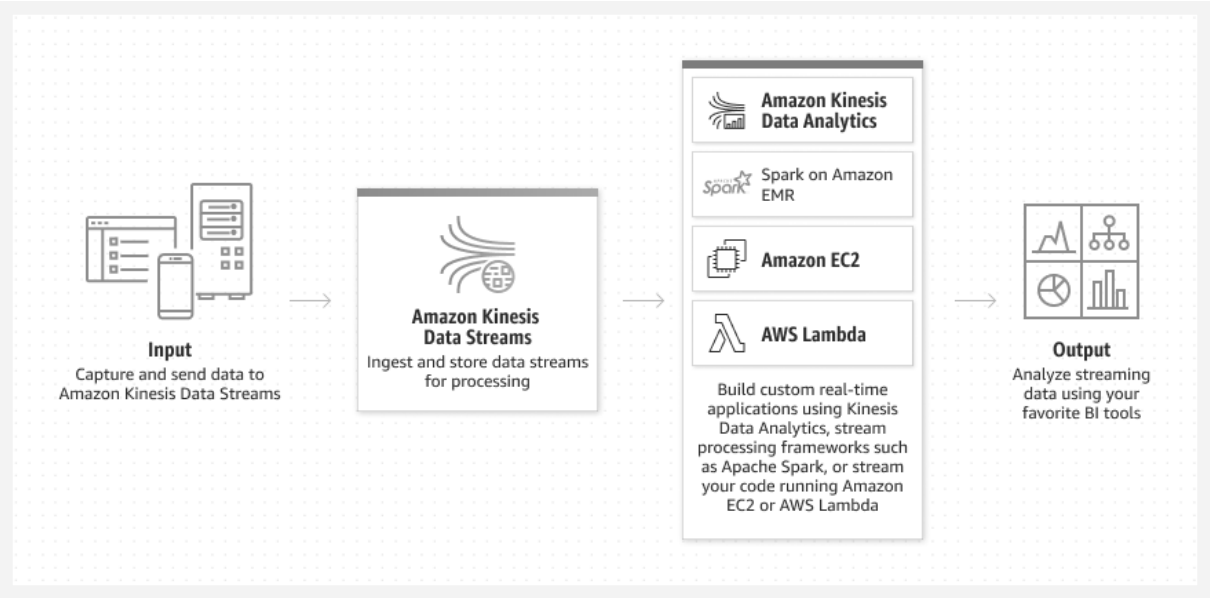



Рисунок 1.8 – Загальна схема роботи Amazon Kinesis

**Amazon Simple Storage Service (Amazon S3)** – це сервіс об’єктного сховища, що надає масштабованість, доступність даних, безпеку та продуктивність. Замовники будь-якого класу та будь-якої індустрії можуть зберігати та захищати будь-яку кількість інформації віртуально для будь-якого випадку, такого як озеро даних (data lake), хмарних додатків та мобільних додатків [49]. Послуга надає можливість оптимізувати кошти за допомогою класів зберігання (наприклад сервіс

може забезпечити стійкість до втратити даних з 99,9999999999 %), а за допомогою простими у користуванні функціями – організація інформації та налаштування контролів доступу за бізнес, організаційними або відповідними вимогами (рис 1.9).

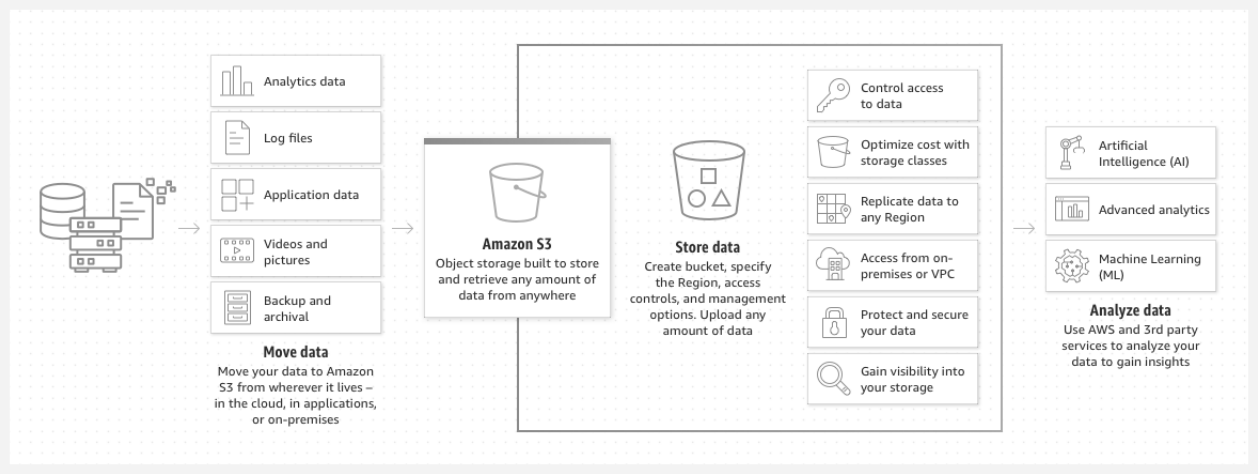



Рисунок 1.9 – Загальна схема Amazon S3

**AWS Glue Data Catalog** – використовується для швидкого виявлення та пошуку багатьох AWS наборів даних без переміщення самої інформації (рис 1.10).


Рисунок 1.10 – Загальна схема AWS Glue Data Catalog

Після індексації даних, вони одразу доступні для пошуку та запитів за допомогою Amazon Athena, Amazon EMR та Amazon Redshift Spectrum [56].

**Amazon Athena** – безсерверний інтерактивний аналітичний сервіс, побудований на базі фреймворків з відкритим кодом та підтримкою форматів відкритих таблиц та файлів (рис 1.11). Athena надає спрощений та гнучкий спосіб аналізу петабайтів інформації там, де вони знаходяться. Аналізуйте дані і створюйте додатки на основі озера даних (data lake) Amazon Simple Storage Service (S3) та понад 30 джерел даних, у тому числі локальні джерела інформації або інші хмарні системи, використовуючи SQL або Python. Athena побудована на двигунах Trino та Presto з відкритим кодом и фреймворке Apache Spark. Даний сервіс не передюачує ніяких зусиль по забезпеченню або налаштуванню [57].

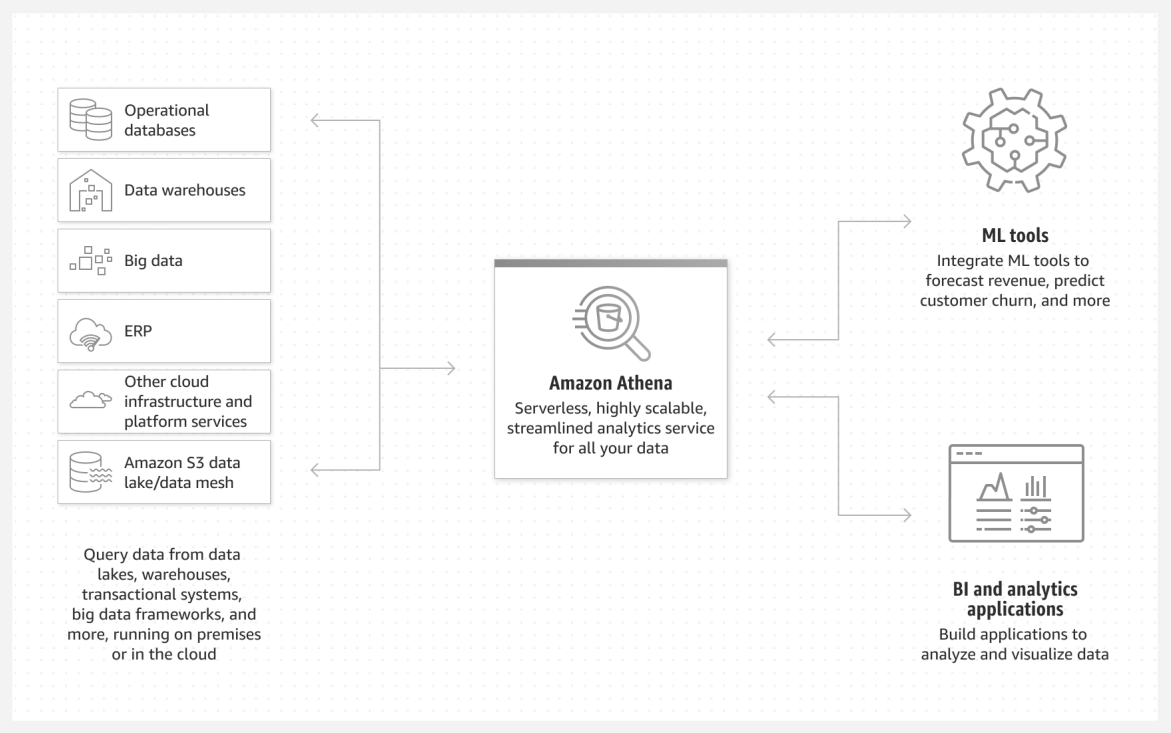



Рисунок 1.11 – Загальна схема AWS Glue Data Catalog

**Amazon SageMaker** – це повністю підконтрольний сервіс, що об’єднує набір інструментів для забезпечення високопродуктивного та недорогого машинного навчання (ML) при любому сценарії застосування [50]. SageMaker надає можливість створювати, навчати та розгортати моделі машинного навчання необхідного масштабу, використовуючи такі інструменти як блокноти, відладчики, профільувальники, конвеєри - і все це в одному інтегрованому середовищі

розробки (IDE). Сервіс підтримує вимоги до керування інформацією, забезпечуючи спрощений контроль доступу та підвищену прозорість ваших проектів машинного навчання. Крім цього є можливість створювати свої базові моделі, а також великі моделі, які навчалися на великих масивах даних, за допомогою спеціальних інструментів для точного налаштування, експериментального вивчення, перенавчання та розгортки базових моделей. Дана послуга надає доступ до сотні заздалегідь навчених моделей, до яких належать загальнодоступні базові моделі, що можна розвернути двома кліками миші.

**Amazon SageMaker Data Wrangler** зменшує час для агрегації та підготовки табличних даних та зображень для машинного навчання з неділь до хвилин (рис 1.12) [55,58].

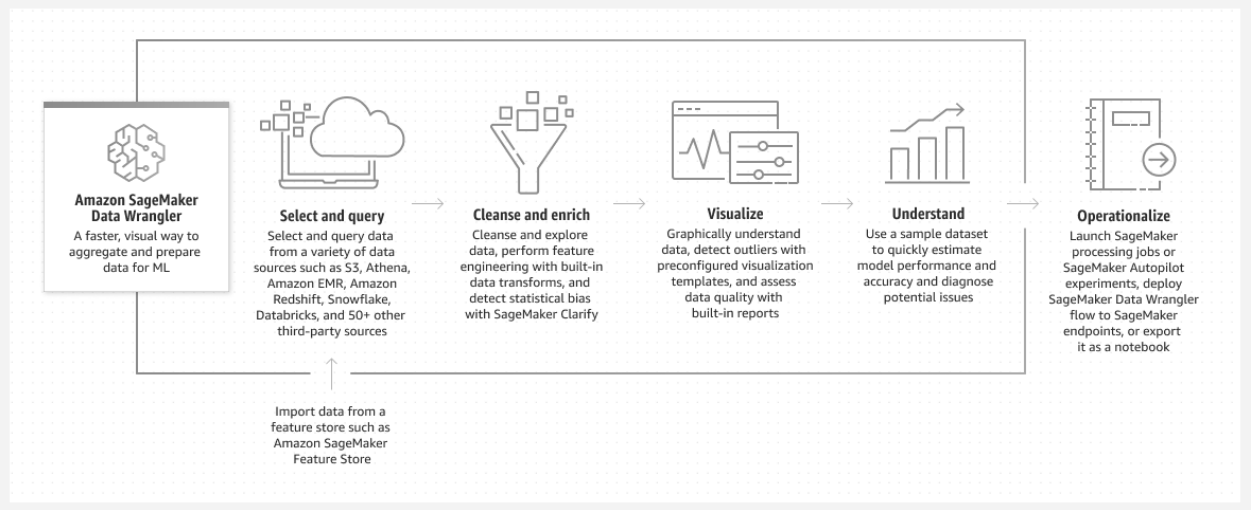



Рисунок 1.12 – Amazon SageMaker Data Wrangler

Сервіс надає можливість спрощення процесу підготовки даних та функцій. Єдиний графічний інтерфейс Data Wrangler дозволяє закінчити кожний крок схеми підготовки даних, в яку входить вибір інформації, очищення, дослідження, візуалізація та масштабування обробки. Є можливість використання SQL для створення вибірки з різних джерел даних та швидкого імпорту. Якість даних та аналітичні звіти автоматично перевіряють інформацію та виявляють аномалії, такі як дублікати та протікання цілі. SageMaker Data Wrangler має 300 способів

вбудованої трансформації даних, що дозволяє перебудувати інформацію без написання коду.

**Amazon SageMaker Feature Store** – це повністю керований, спеціально створене сховище для зберігання, обміну та керування функціями для моделей машинного навчання (ML). Функції це вхідні дані для ML моделей, які використовувались під час навчання та умововиведення. Наприклад, у додатку, що надає рекомендації стосовно музичного списку, до функцій можуть входити рейтинг пісні, її тривалість та демографічні висновки щодо слухачів. Функції використовуються неодноразово багатьма командами. Якість функцій має критичне значення для забезпечення високої точності моделі. Також, коли функції використовуються для тренування моделей поступово в автономному режимі, стають доступними для виявлення умововиведення у реальному часі, то важко зберігати дві функції синхронізованими. SageMaker Feature Store надає захищене та уніфіковане сховище для обробки, стандартизації та використання функцій у масштабі протягом життєвого циклу ML (рис 1.13) [50].

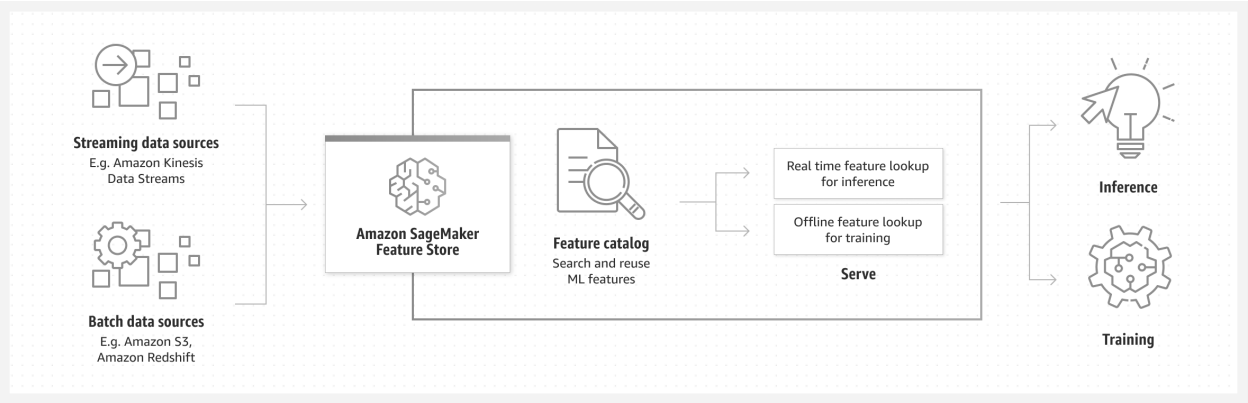



Рисунок 1.13 – Загальна схема Amazon SageMaker Feature Store

**Amazon SageMaker Model Registry** дозволяє виконувати наступні дії:

- Створювати каталог моделей для продуктивного середовища
- Керувати версіями моделей
- Асоціювати метадані, такі як тренувальні метрики, з моделю
- Керувати статусом схвалення моделі

- Розгорнути моделі на продуктивне середовище
- Автоматизувати розгортку моделі за допомогою CI/CD

Каталоги моделей створені за допомогою Amazon SageMaker Model Registry групи (пакети) моделей, які містять різні версії моделі. Користувач може створити групу моделей, яка відслідковує усі моделі, які користувач натренував для вирішення проблеми. Є можливість рєстрація кожної моделі, яку тренує користувач, та Model Registry додасть її у групу моделей як нову версію моделі. Під кінець користувач може створити категорії груп моделей для подальшої організації їх у SageMaker Model Registry Collections [61].

**Amazon SageMaker Model Monitor** – моніторить якість Amazon SageMaker ML моделі на продуктивному середовищі. Користувач може побудувати безперервний моніторинг за допомогою кінцевих точок у режимі реального часу (або групи задач перетворення, що регулярно запускаються) або моніторинг по розкладу для асинхронної групи задача перетворення. За допомогою даного функціоналу, користувач може налаштувати нотифікації по аваріям, коли є розбіжності у якості моделі. Раннє та проактивне виявлення цих розбіжностей надає можливість зробити корекційні дії, такі як перетренування моделі, аудит вищих систем або виправлення проблем якості без необхідності ручного моніторинга моделі або створення додаткових інструментів. Є можливість використання вбудованого функціоналу моніторингу, який не вимагає програмування. Або користувач може побудувати свій моніторинг за допомогою програмування, щоб налаштувати свою аналітику [61].

**Кінцеві точки Amazon SageMaker** – після розгортання моделі на кінцеву точку, у користувача є можливість дивитися її статус та керувати нею. За допомогою SageMaker, користувач має можливість дивитися статус та деталі кінцевої точки, перевіряти метрики за журнали записів для моніторингу продуктивності кінцевої точки, оновлювати розгорнуту модель на кінцевій точці, тощо [59].

**AWS Lambda** – це обчислювальний сервіс, який запускає код у відповідь на подію та автоматично керує обчислювальними ресурсами, що дозволяє швидше

перетворити ідею на сучасні виробничі безсерверні додатки (рис 1.14). До списку подій входять зміни в стані або оновлення, наприклад, коли користувач додає товар в корзину на веб-сайті інтернет-комерції. AWS Lambda також використовується для розширення можливостей інших послуг AWS за допомогою спеціальної логіки або для створення своїх серверних сервісів с використанням можливостей масштабування, продуктивності та безпеки AWS. Як описано вище, сервіс автоматично запускає програмний в відповідь на різні події, такі як запити HTTP через Amazon API Gateway, зміни об'єктів в корзині Amazon Simple Storage Service (Amazon S3), оновлення таблиц в Amazon DynamoDB або зміна стану в AWS Step Functions [62, 63].

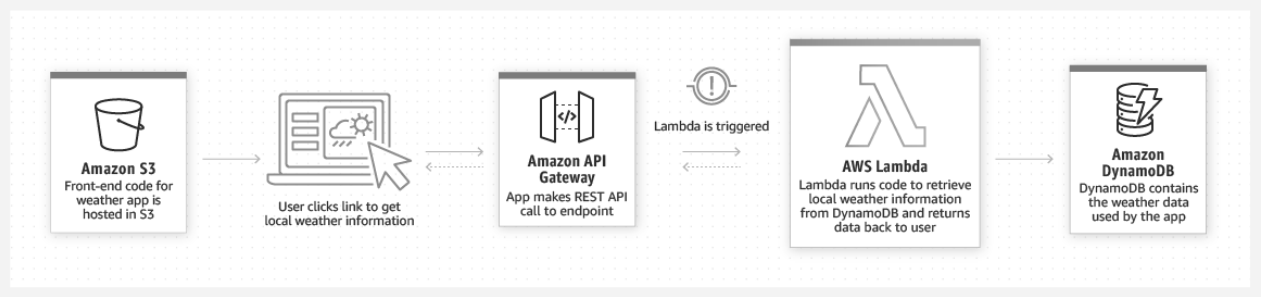



Рисунок 1.14 – Загальна схема AWS Lambda для Web додатків

**Amazon API Gateway** – це повністю керуємий сервіс для розробників, призначений для створення, публікації, обслуговування, моніторингу та забезпечення безпеки API в будь-яких масштабах. Через API додатку отримують доступ до даних , бізнес-логіці або функціональним можливостям ваших серверних сервісів. Дана послуга дозволяє створювати API RESTful и WebSocket, які є головним компонентом додатків для двостороннього зв'язку у режимі реального часу. API Gateway підтримує робоче навантаження в контейнерах та безсерверні робочі навантаження, а також інтернет додатки[53].

API Gateway бере на себе усі задачі, які пов'язані с прийомом та обробкою сотні тисяч одночасних викликів API, в тому числі керування трафіком, підтримка CORS, автоматизацію та контроль доступу, регулювання кількості запитів,

моніторинг та керування версіями API. Робота з даним сервісом не потребує мінімальних затрат або стартових вкладень (рис 1.15).

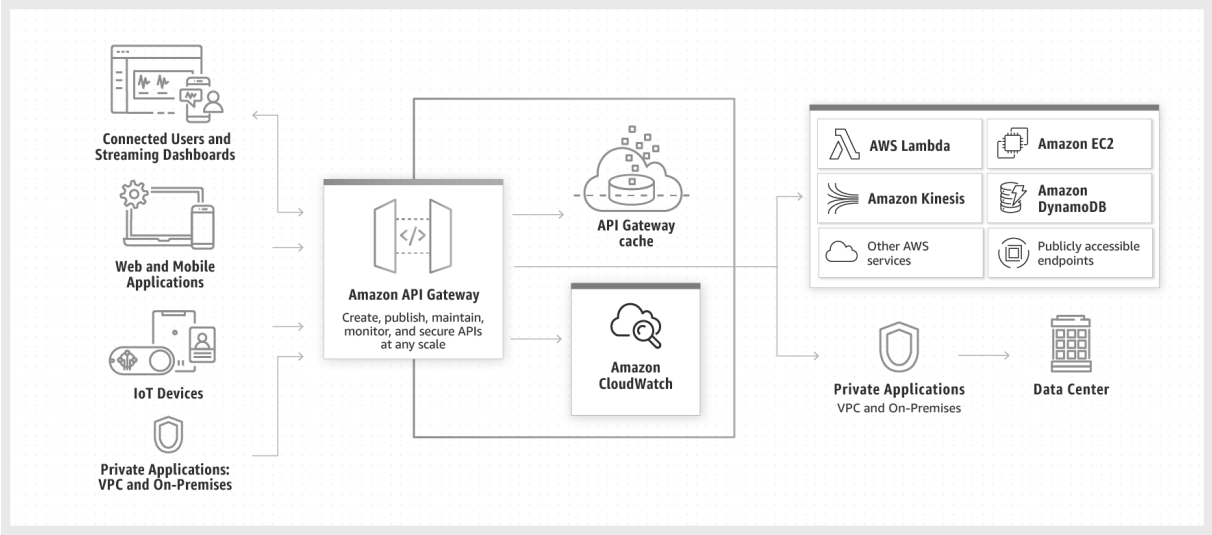



Рисунок 1.15 – Загальна схема Amazon API Gateway

Користувач платить за отримані виклики API та переданий об’єм даних і може за допомогою багаторівневої моделі ціноутворення API Gateway знизити свої витрати по мірі масштабування користування API.

### 1.2.2 Огляд складових Hadoop

Величезна кількість зібраних даних має потенціал для виявлення корисних тенденцій та моделей. Звідси треба зберігати та обробляти. Всі ці дані зберігаються в хмарі або великих вторинних сховищах, таких як файлова система Hadoop (Hadoop Distributed File System - HDFS). Обробка здійснюється за допомогою Hadoop або Spark. Основна думка з аналізу цих даних - це численні програми для бізнес-аналізу, виявлення шахрайства, прогноз погоди, персоналізована реклама тощо. Для аналізу такого роду використовуються різні інструменти для машинного навчання та інструменти для обробки даних.

Нижче наведено деякі технології, які допомагають користувачам ефективно обробляти та обробляти великі дані. Велика обробка даних може бути зроблена з урахуванням наступних аспектів:

- Обробка великих даних: MapReduce, Hadoop є інтегрованою структурою для обробки та зберігання Великих даних
- Аналіз та запит даних: WibiData, PLATFORA, PIG
- Бізнес-аналітика: Hive
- Зберігання: хмарне сховище, колоночна база даних (Column-Oriented Databases) , безсхемна база даних (Schema-Less Databases або NoSQL Databases)
- Машинне навчання: Apache Mahout, SkyTree

Деякі з різних методів обробки великих даних наведені нижче [64]:

**MapReduce** - це ключовий алгоритм, який використовує двигун Hadoop MapReduce для розподілу роботи у кластері.

Функція Mapper - Функція перетворення карти передбачена для перетворення рядка вхідних даних клавіші та значення у вихідну клавішу / значення:

- map(key1,value) -> list<key2,value2>

Тобто для введення він повертає список, що містить нуль або більше (ключ, значення) пар: вихідне значення може бути іншим ключем від введення. Воно може мати кілька записів з однаковими клавішами.

Скорочення функції: передбачено перетворення зменшення, щоб взяти всі значення для певної клавіші та створити новий список зменшеного виходу.

- reduce(key2, list<value2>) -> list<value3>

**Apache Hadoop** - це платформа з відкритим кодом для розподіленого зберігання та обробки великих наборів даних на товарній техніці. Hadoop дає компаніям можливість швидко отримувати інформацію від великої кількості структурованих та неструктурованих даних. Він використовується для підтримки, масштабування та аналізу великого обсягу даних. Ці дані можуть бути структурованими або неструктурованими.

**Apache PIG** - це платформа для аналізу великих наборів даних. Мова PIG, PIG Latin, дозволяє вказати послідовність функцій перетворення, таких як об'єднання, фільтрація, групування тощо. Окрім вбудованих функцій, він також забезпечує

можливість користувальницьких функцій виконувати спеціальну обробку. Мова PIG дозволяє виконувати запити за даними, що зберігаються на кластері Hadoop, а не на "SQL-подібну" мову.

**Hive** дозволяє традиційним додаткам BI запускати запити проти кластера Hadoop. Вона була розроблена спочатку компанією Facebook, але вже протягом деякого часу вона стала відкритою, і є абстракцією високого рівня структури Hadoop, що дозволяє хто б не робив запити щодо даних, що зберігаються в кластері Hadoop, так, якби вони маніпулювали звичайним сховищем даних. Це робить Hadoop більш корисним для користувачів BI.

**Column-Oriented Databases** – це звичайні колонкові бази даних, які найкраще підходять для обробки онлайн-транзакцій з високою швидкістю оновлення, але вони не відповідають ефективності запитів, оскільки обсяги даних зростають, а дані стають більш неструктурованими. Column-Oriented Databases зберігають дані з фокусом на стовпці, а не на рядки, що дозволяє здійснювати величезне стиснення даних і дуже швидке запит. До таких БД належать Greenplum та Sybase IQ.

**Schema-Less Databases або NoSQL Databases** - у цю категорію входять декілька типів баз даних, таких як сховища ключових значень та сховища документів, які зосереджують увагу на зберіганні та вилученні великих обсягів неструктурованих, напівструктурованих або навіть структурованих даних. Вони досягають виграшу продуктивності, відміняючи деякі (або всі) обмеження, традиційно пов'язані з звичайними базами даних, такі як узгодженість читання-запису, в обмін на масштабованість та розподілену обробку.

**Хмарні сховища.** В склад Hadoop входить система розподіленого сховища Hadoop Distributed File System (HDFS), яка дозволяє зберігати дані в великих блоках на локальних дисках кластеру. HDFS пропонує налаштовуваний коефіцієнт реплікації (за замовчуванням 3x) за рахунок якого забезпечується підвищена доступність та надійність. HDFS реалізує моніторинг реплікацій та балансування даних між вузлами у міру виходу вузлів з ладу та додавання нових вузлів. HDFS також можна використовувати для зберігання вхідних і вихідних даних в сервісі Amazon S3.

Завдяки файловій системі EMR File System (EMRFS) кластерів Amazon EMR стає можливим використання послуги Amazon S3 в якості рівня зберігання для Hadoop. За рахунок зберігання інформації в Amazon S3 є можливість відділення рівня розрахунку/обробки від рівня зберігання, що дозволяє задати розмір кластеру Amazon EMR з розрахунком необхідного об’єму ресурсів ЦП (центрального процесору) та пам’яті для обробки робочих навантажень та виключення з структури кластера надмірні вузли, призначені для максимізації кластерного сховища. Файлова система EMRFS оптимізована під Hadoop та дозволяє ефективно здійснювати паралельні операції читання та запису даних з сервісом Amazon S3, а також обробляти об’єкти, закодовані за допомогою клієнтського та серверного кодування Amazon S3. EMRFS дозволяє використовувати Amazon S3 в якості озера даних, при цьому Hadoop в Amazon EMR є можливість використовувати в якості рівня еластичних запитів [52].

Hadoop може бути розділений на 4 функціональні рівні (рис 1.16)

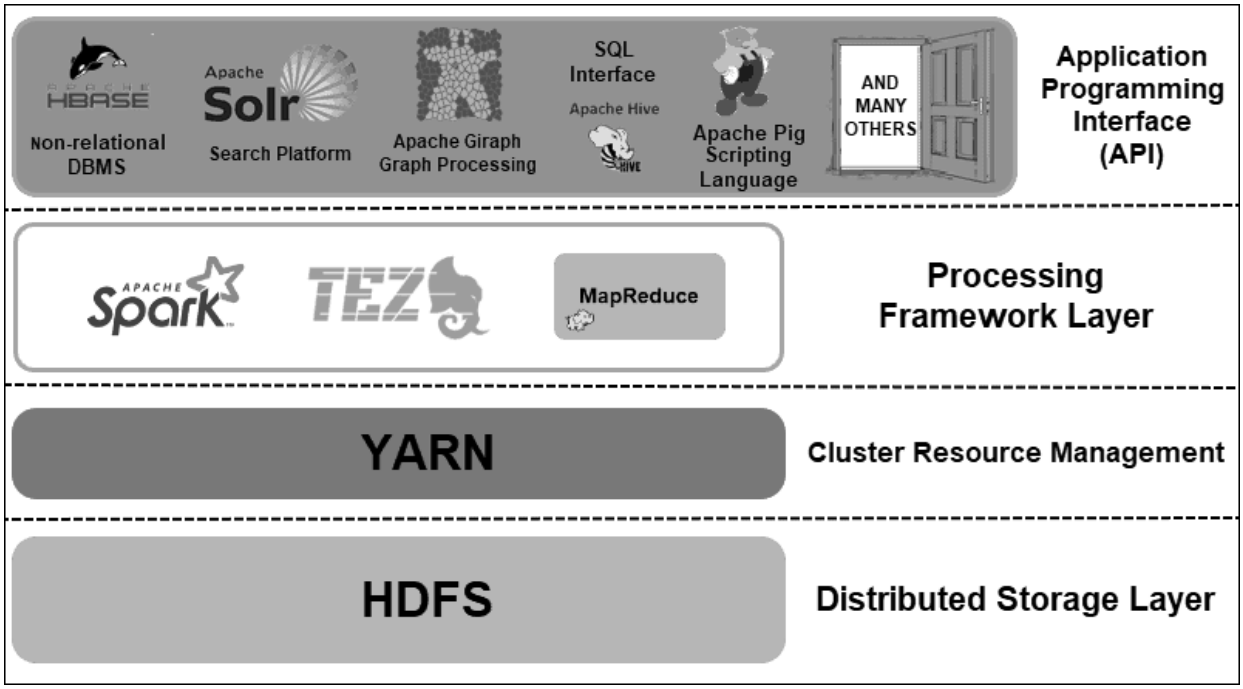



Рисунок 1.16 – Функціональні рівні Hadoop

Перший рівень - це рівень розподіленого сховища (Distributed Storage Layer). Кожна вузол в Hadoop кластері має свій власний дисковий простір, пам'ять,

пропускну спроможність та обробку. Дані, які надходять до системи, розбиваються на індивідуальні юлоки даних, що зберігаються в межах рівня розподіленого сховища HDFS. HDFS передбачає, що кожен дисковий масив та підконтрольний вузол в межах кластеру не є надійним. Тому, як запобіжний захід, HDFS зберігає три копії кожного набору даних в межах кластеру. Головний вузол HDFS (NameNode) зберігає метадані для індивідуальних блоків інформації та всіх їх копій.

Другий рівень – це рівень керування ресурсами кластеру. Hadoop необхідно ідеально координувати вузли, щоб нечисленна додатків та користувачів могли ефективно розподіляти свої ресурси. С самого початку, MapReduce виконував функції керування ресурсами та обробкою даних. YARN розділяє ці дві функції. Як де-факто інструмент для керування ресурсами Hadoop, YARN зараз може виділяти ресурси для різних фреймворків написаних під Hadoop. До цих фреймворків входять Apache Pig, Hive, Giraph, Zookeeper та MapReduce.

Третій рівень – це рівень обробки фреймворками. Даний рівень обробки складається з фреймворків, що аналізують та оброблюють. Структуровані та неструктуровані набори даних проходять процеси відображення, перетасування, сортування, об’єднання та зменшення у маленькі керовані блоки інформації. Ці операції поширені серед багатьох вузлів як можна ближче до серверів, де знаходиться інформація. Фрейворки обчислення такі як Spark, Storm, Tez надають можливість обробки у режимі реального часу, інтерактивну обробку запитів та інші програмні опції які допомагають двигуну MapReduce та утилізують HDFS набагато ефективніше.

Четвертий рівень – інтерфейс програмування додатків. Введення YARN у Hadoop 2 дозволили створити нові фреймворки обробки та API. Big data продовжує розширення та різноманітність інструментів повинна слідувати за цим зростанням. Проекти, що націлені на пошукові платформи, потокові дані, зручні для користувача інтерфейси, мови програмування, обмін повідомленнями, відмовостійкість та безпеку – всі вони є маленькими частинами комплексної екосистеми Hadoop[51].

### 1.2.3 Складові HDFS та принцип роботи його елементів

Розглядаючи архітектуру HDFS можна помітити що він є відмовостійким по дизайну. Як було зазначено до цього, інформація зберігається в індивідуальних блоках даних у три розділені копії поміж різних вузлів. HDS складається з двох типів елементів – DataNode та NameNode.

**DataNode** оброблюють та зберігають блоки даних, а тим часом NameNode керують DataNode, керуючи метаданими блоками даних, та контроль над клієнтським доступом (рис. 1.17).

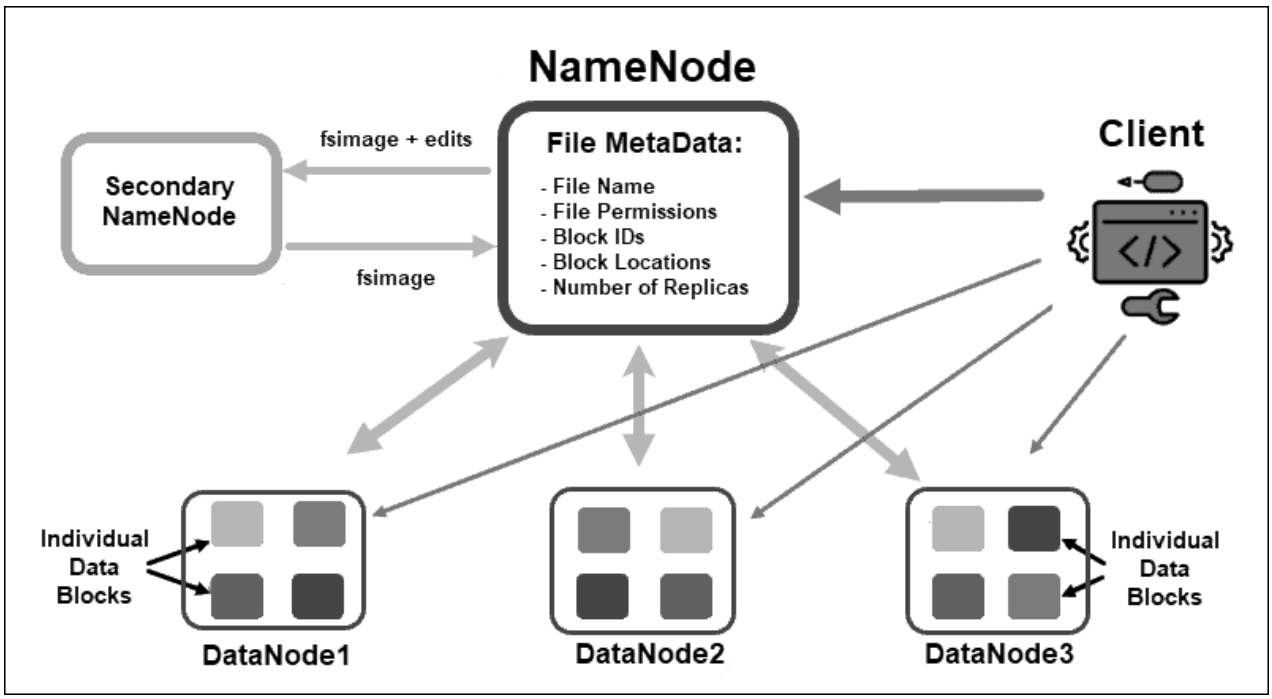



Рисунок 1.17 – Схема взаємодії елементів HDFS

**NameNode** має наступні властивості – спочатку, інформація розбиваються в абстрактні блоки даних. Метадані файлів для цих блоків, в які містять в собі ім’я файлу, дозволи до файлу, ІД, місцезнаходження, кількість копій, зберігаються у fsimage, що розташований на локальній пам’яті NameNode.

Якщо NameNode виходить з ладу, HDFS не зможе визначити місцезнаходження будь-якого набору інформації розподіленої поміж DataNode. Іншими словами, NameNode – це єдина точна відмови всього кластеру. Ця

вразливість може бути урегульована встановленням вторинної NameNode або іншої NameNode, що знаходиться у режимі очікування.

У більш ранніх версіях Hadoop вторинна NameNode слугувала як основне рішення резервного копіювання. Цей елемент час від часу завантажував поточний fsimage та редагував журнали роботи з NameNode та об'єднував їх. З цього відредагованого fsimage головна NameNode може бути відновлена.

У цьому сценарії резервування процес не є автоматизованим, тому що адміністратору необхідного буде відновити дані з вторинної NameNode власноруч.

Функція високої доступності була інтегрована у Hadoop 2.0 та наступних версіях, щоб запобігти будь-якого часу простою у випадку відмови NameNode. Це дозволяє керувати двома NameNode, що працюють на різних відділених головних вузлах.

NameNode, що знаходиться у режимі очікування, автоматично переходить до роботи, у випадку якщо активна NameNode дає відмову у роботі. Цей елемент також виконує функцію контрольно-пропускного процесу. Завдяки цій функції, вторинна NameNode та NameNode у режимі очікування не є сумісними. Hadoop кластер може мати лише одну з них.

**Zookeeper** – це інструмент що підтримує високу доступність та резервування. Цей інструмент не використовує багато ресурсів системний та швидкий у користуванні. NameNode у режимі очікування підтримує активний сеанс з демоном Zookeeper.

Якщо активна NameNode дає збій, демон Zookeeper виявляє помилку та здійснює процес перенесення функцій на нову NameNode. Використання Zookeeper автоматизує процес резервування та мінімізує вплив відмови у роботі NameNode на кластер.

Кожна DataNode у кластері використовує процес на задньому плані для зберігання індивідуальних блоків даних на підлеглих серверах.

За замовчуванням, HDFS зберігає три копії кожного блоку даних на різних DataNode. NameNode використовує політику розміщення з урахуванням серверної

шафи (rack-aware placement policy). Це означає, що DataNode які містять копії блоків даних не можуть бути розташовані на одній серверній стійці.

DataNode підтримують зв’язок та отримують інструкції від NameNode десь 20 раз за хвилину. Також воно передає стан блоків даних на вузлі один раз на годину. На базі отриманої інформації, NameNode може відправити запит на створення додаткових копій, видалити їх, або зменшити кількість блоків даних, що знаходяться на вузлі.

**Rack-aware placement policy** – це одне з головних визначень розподілених систем зберігання як HDFS, а саме підтримка високої доступності та копіювання. Тому, блоки даних повинні бути розподілені не тільки на різних DataNode, а й на вузлах розташованих на різних серверних шафах (рис. 1.18).

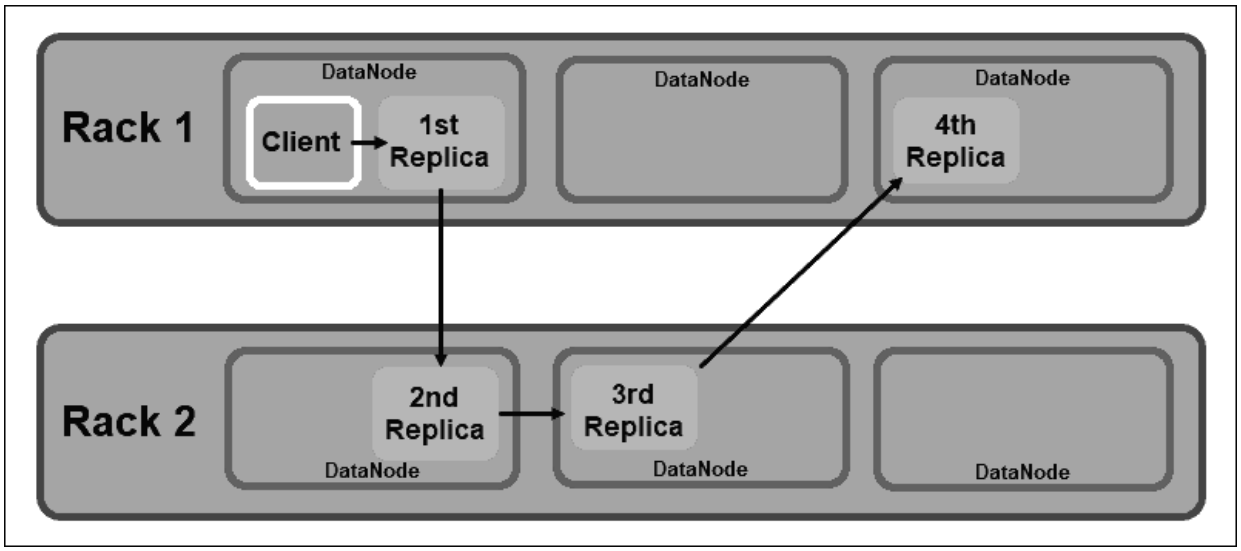



Рисунок 1.18 – Схема взаємодії елементів HDFS

Це забезпечує, що при випадку при відмові у роботі усієї серверної шафи, не будуть знищені усі копії інформації. HDFS NameNode забезпечує наступну політику розміщення з урахуванням шафи (rack-aware placement policy):

- Перша копія блоку даних розташовується на тому ж самому вузлі, що й клієнт;
- Друга копія автоматично розташовується на випадковому DataNode в іншій шафі;

- Третя копія розмішується на іншій DataNode в одній шафі, де знаходиться друга копія;
- Будь-які додаткові копії зберігаються на випадкового розподілених DataNode у кластері;

Політика розміщення з урахуванням шафи утримує лише одну копію на вузлі та встановлює ліміт до двох копій в одній серверній шафі.

Відмова у роботі шаф стається рідше ніж відмова вузлів. HDFS забезпечує високу доступність при постійному зберіганні хоча б однієї копії блоку даних у DataNode, розташованому у іншій шафі [51].

### 1.2.4 Складові YARN та принцип роботи його елементів

**YARN (Yet Another Resource Negotiator)** це стандартний елемент керування ресурсами кластеру для Hadoop 2 та Hadoop 3 (рис 1.19).

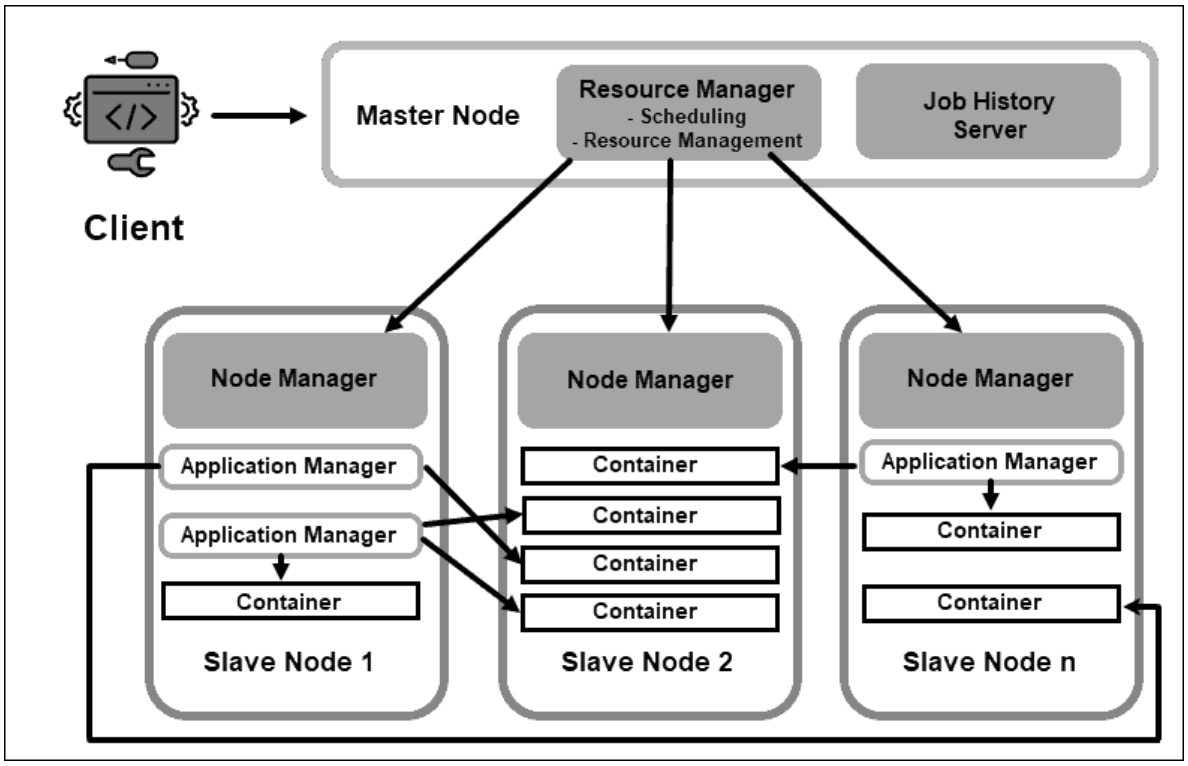



Рисунок 1.19 – Взаємодія YARN з елементами Hadoop

У минулих версіях Hadoop використовувався MapReduce для керування обробкою даних та виділенням ресурсів. З плином часу, необхідність розділення керування ресурсами та обробкою стало передумовою розробки YARN.

Роль YARN як елемента керування ресурсами встановлює її між рівнем зберіганням, представленим HDFS, та двигуном обробки MapReduce. YARN також надає загальний інтерфейс, що дозволяє користувачу додавати нові двигуни обробки для різних типів даних.

**Демон ResourceManager (RM)** контролює усі ресурси обробки у Hadoop кластері. Основне призначення RM полягає у призначенні ресурсів індивідуальним додаткам, розташованим на підконтрольних вузлах. Воно контролює глобальний огляд запланованих та тривалих процесів, опрацьовує запити на ресурси, планує та виділяє ресурси відповідно. ResourceManager є необхідним для фреймворку Hadoop та повинен працювати на виділеному керуючому вузлу.

Єдиний фокус RM це планування навантаження. На відміну від MapReduce, цей демон не виконує функції відновлення після відмови або індивідуальні задачі обробки. Розділення задач у YARN є тим, що робить Hadoop за своєю суттю масштабованим та перетворює його в повністю розроблену комп’ютерну платформу.

Кожна підконтрольний вузол має сервіс обробки NodeManager та сервіс зберігання DataNode. Разом вони формують основу розподіленої системи Hadoop.

DataNode, як було зазначено вище, є елементом HDFS та контролюється NameNode. NameNode, в схожому сенсі, підконтрольний RM. Головна функція демона NodeManager полягає у відстеженні даних обробки та ресурсів на підконтрольному вузлу та відправлені періодичних звітів до ResourceManager.

YARN демони та контейнери є Java процесами, що працюють у Java VM.

Ресурси обробки в Hadoop завжди надаються в контейнерах. Контейнер має пам'ять, системні файли та простір для обробки.

Розвертка контейнеру є загальна та може бути запустити будь-який ресурс на запит у будь-якій системі. Якщо кількість запрошеного ресурсу кластеру знаходиться у рамках прийнятного ліміту, RM приймає запит та запланує контейнер для розгортання.

Процеси контейнеру на підконтрольному вузлу спочатку надаються, перевіряються та відстежуються NodeManager на заданій підконтрольному вузлу.

Кожен контейнер на підконтрольному вузлу має свій виділений Application Master. Application Master розгортаються у вигляді контейнера. Навіть MapReduce має Application Master, що виконує функції відображення та зменшення.

Доки Application Master є активним, він відправляє повідомлення до Resource Manager щодо його теперішнього стану та стану додатку, що він моніторить. На базі наданої інформації, Resource Manager може запланувати додаткові ресурси або зазначити їх кудись в кластері, якщо ці ресурси не потрібні.

**Application Master** наглядає за повним життєвим циклом додатка, весь його шлях починаючи від запиту на необхідні контейнери від RM до подання запиту на оренду контейнеру до NodeManager.

**JobHistory Server** дозволяє користувачам отримати інформацію стосовно додатків, що закінчили свою роботу. REST API надає функціональну сумісність та може динамічно надати інформацію користувачам щодо поточних та закінчених роботах виконаних сервером.

Базовий робочий процес на розгортання у YARN починається, коли клієнтський додаток надає запит до Resource Manager.

1) Resource Manager інструктує NodeManager запустити Application Master для цього запиту, що потім розпочинається в контейнеру.
2) Новостворений Application Master реєструє самого себе за допомогою RM. Потім Application Master починає сеанс зв’язку з HDFS NameNode та визначає місцезнаходження необхідних блоків даних та розраховує кількість необхідних задач відображення та зменшення для обробки інформації.
3) Application Master надсилає запит на потрібні ресурси на RM та продовжує відслідковувати потреби на ресурси протягом життєвого циклу контейнеру.
4) RM запланує виділення ресурсів з урахуванням запитів від усіх інших Application Master та встановлює в чергу їх запити. Після того як потужності звільняються, RM робить їх наявними для Application Master на конкретному підконтрольному вузлу.

5) Application Master контактує з NodeManager стосовно підконтрольному йому вузлу та робить запит на створення контейнеру. У цьому запиті міститься інформація стосовно змінних, токенів аутентифікації та ланцюжок команд для цього процесу. На базі цього запиту NodeManager створює та вмикає контейнер у роботі.
6) Потім Application Manager наглядає за станом процесу та реагує на події відмов роблячи перезапуск процесу у наступному вільному слоті. Якщо відмови виявляються після чотирьох різних спроб, то вся задача дає відмову. Упродовж всього процесу, Application Manager надсилає відповіді на запити стану від клієнта.

Після того, як усі завдання були виконані, Application Master надсилає результат на клієнтській додаток, інформуючи RM, що додаток закінчив своє завдання, проводить зняття з реєстрації у Resource Manager та сам себе вимикає.

RM також надає інструкції NameNode для видалення конкретного контейнеру під час процесу у випадку зміни пріоритету обробки [51].

### 1.2.5 Складові MapReduce та принцип роботи його елементів

MapReduce це програмний алгоритм обробки даних розсіяних по Hadoop кластеру. Як і будь-який інший процес в Hadoop, коли стартує задача MapReduce, Resource Manager робить запит для Application Master на керування та моніторинг життєвого циклу задачі MapReduce.

Application Master знаходить необхідні блоки даних за допомогою інформації, що зберігається в NameNode. AM також інформує ResourceManager розпочати задачу MapReduce на тих самих вузлах, де знаходяться блоки даних. По можливості, інформація оброблюється локально на підконтрольних вузлах щоб зменшити використання пропускної спроможності та підвищити ефективність кластеру.

Вхідні дані проходять **відображення**, **перемішування** та потім **стиснення (зменшення)** до агрегованого результату. Результат задачі MapReduce зберігається та копіюється в HDFS.

Hadoop сервера, що виконують функції відображення та зменшення частіше за всього називають **Mapper** та **Reducer** (рис 1.20).

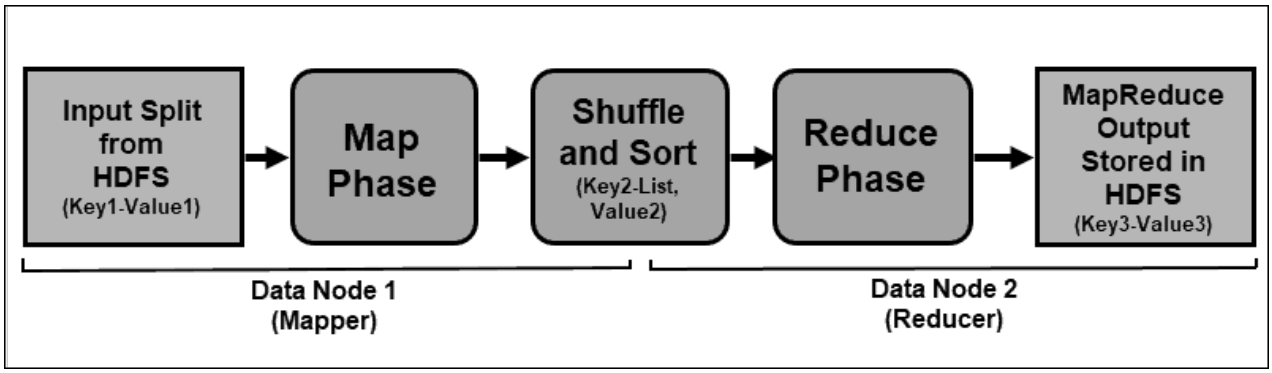



Рисунок 1.20 – Принцип роботи MapReduce

ResourceManager вирішує скільки mapper необхідно використовувати потужностей. Це рішення залежить від розміру оброблених даних та блоків пам'яті, що є в наявності на кожному mapper сервері.

**Процес відображення** поглинає окремі логічні вирази даних, що зберігаються в блоках даних HDFS (рис 1.21).

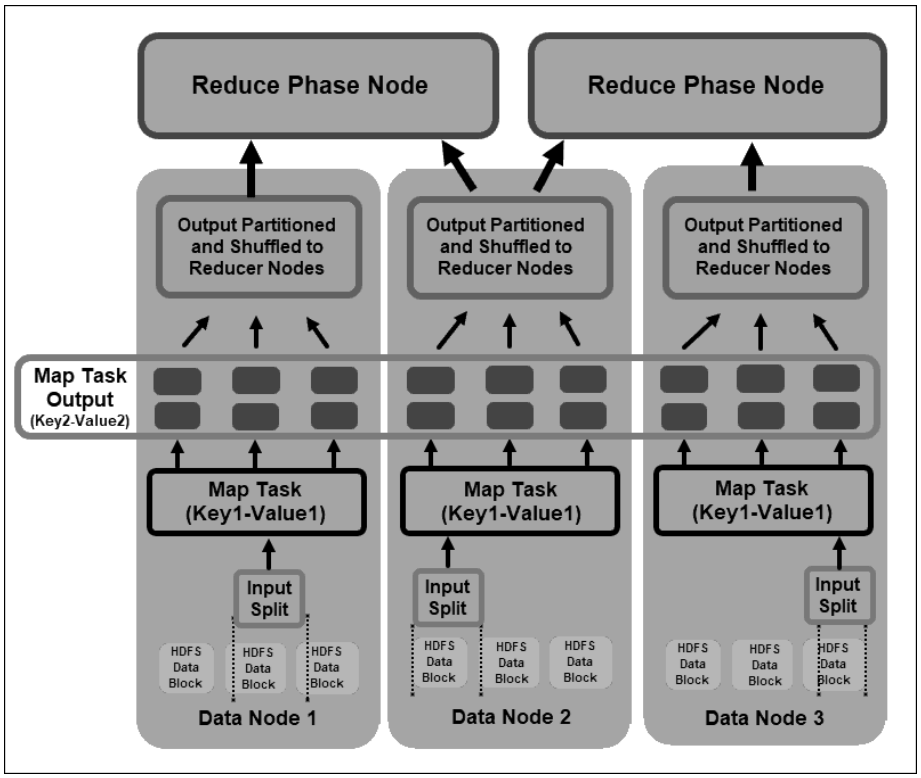



Рисунок 1.21 – Принцип роботи фази відображення

Ці вирази відхватити декілька блоків даних та мають назву input splits. Input splits представлені в процесі відображення як пара ключ-значення.

Задача mapper проходить скрізь кожну пару ключ-значення та створює нову пару ключ-значення, що відрізняється від вхідної пари. Повний асортимент всіх пар ключ-значення представляє вихідний результат роботи задачі mapper.

На основі ключа з кожної пари, дані згруповані, розділені та перемішані до вузлів reducer.

**Перемішування** – це процес, у якому результаті з задачі відображення копіюються до вузлів reducer. Копіювання вихідного результату задачі відображення це тільки обмін даними між вузлами під час усієї роботи MapReduce. Цей результат зберігається на локальному диску вузла mapper, а не в HDFS. Це означає, що ця інформація не підпадає під політику розміщення з урахуванням шафи та процеси копіювання Hadoop.

Вихідний результат задачі відображення необхідно влаштовувати щоб покращити ефективність **фази зменшення (стиснення)** (рис. 1.22).

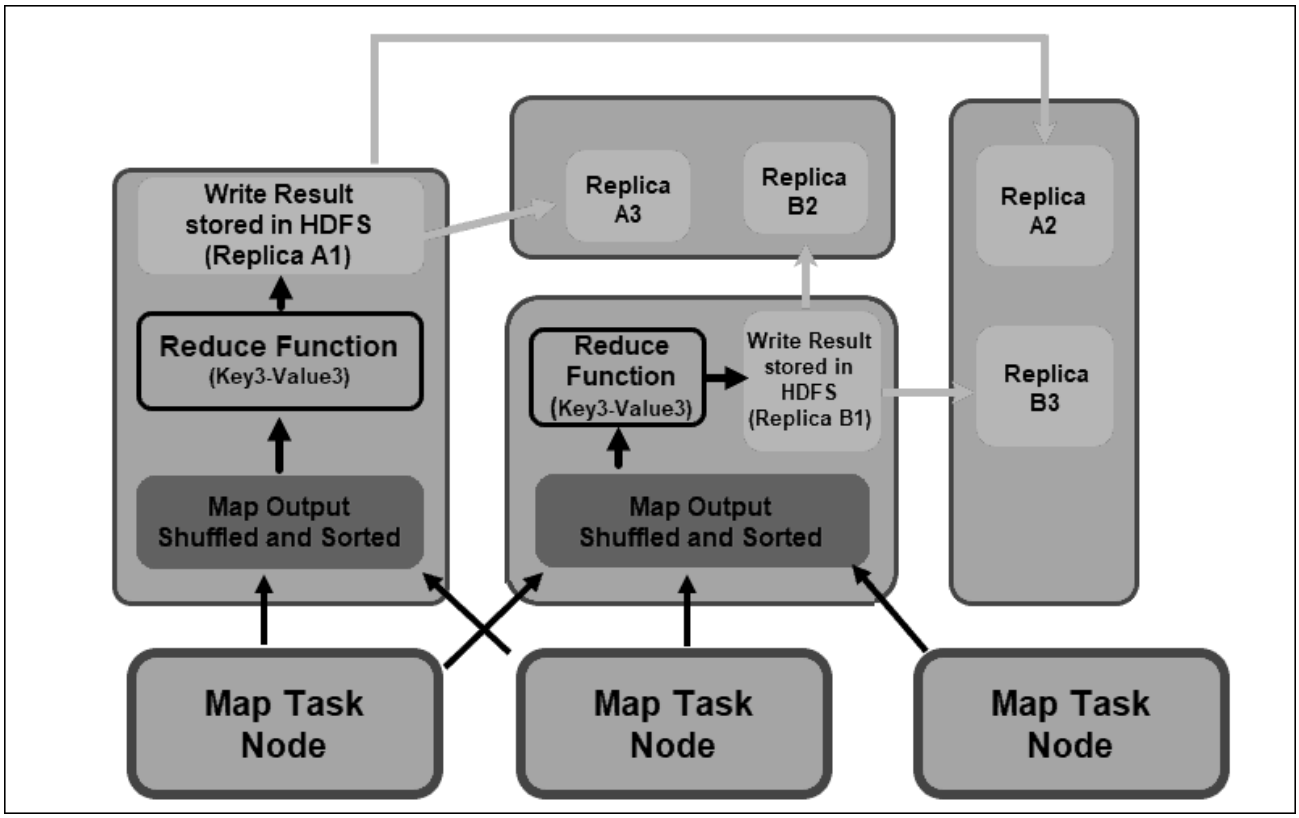



Рисунок 1.22 – Принцип роботи фази стиснення

Відображені пари ключ-значення, перемішані за допомогою вузлів mapper, розташовуються за ключами з відповідними значеннями. Фаза стиснення (зменшення) починається після того, як вхідна інформація просортована по значенню ключа у єдиному вхідному файлі.

Фази перемішування та сортування працюють паралельно. Навіть результати відображення, зібрані з вузлів mapper, групуються та сортуються на вузлах reducer.

Результати відображення перемішуються та сортуються у єдиний стиснений вхідний файл, що знаходиться на вузлу reducer. Функція стиснення (зменшення) використовує вхідний файл для агрегації значень на основі відповідних відображених ключів. Вихідний результат з процесу зменшення – це нова пара ключ-значення. Цей результат є остаточним результатом роботи усього ланцюжка задач MapReduce та, за замовчуванням, зберігається у HDFS.

Всі задачі стиснення працюють одночасно та незалежно одна від одної. Задача зменшення не є обов’язковою.

Також, можуть бути де результат задачі відображення є бажаний результат. У такому випадку необхідність у виведенні єдиного значення не є доцільним [51].

### 1.2.6 Різниця між NoSQL та SQL базами даних

На протязі десятиліть домінуючою моделлю даних при розробці додатків була реляційна модель даних, в якій інформація зберігається в таблицях, які складаються з рядків та стовпців. Для створення та редагування реляційних таблиць використовується структурована мова запитів (Structured Query Language, SQL). Бази даних SQL моделюють відношення між даними у вигляді таблиць. Кожен рядок таблиці являє собою набір взаємопов’язаних значень, що відносяться до одного об’єкту або сутності. Кожен стовпець таблиця є атрибутом даних, а в полі (чи комірці таблиці) зберігаються фактичні значення атрибуту. Можливо використовувати систему керування реляційними базами даних (Relational Database Management System, RDBMS) для доступу до інформації різними способами без реорганізації самих таблиць баз даних.

З середини-кінця 2000го року дедалі частіше стали використовуватися і інші гнучкі моделі даних. Для позначення такого типу баз даних використовують

термінологію «NoSQL». NoSQL позначає «не тільки SQL» або «не SQL». Частіше NoSQL використовують як синонім до нереляційних баз даних. У таблиці наведено відмінність між реляційними та нереляційними базами даних[52].

Таблиця 1.1 – основні відмінності між реляційними та нереляційними базами даних

| | **Реляційні бази даних** | **Бази даних NoSQL** |
|---|---|---|
| Оптимальне робоче навантаження | Реляційні бази даних призначені для трансакційних та наполегливо несуперечливих додатків онлайн обробки транзакцій (OLTP). Вони також гарно підходять для аналітичної онлайн обробки (OLAP). | Бази даних NoSQL призначені для роботи з цілою низкою шаблонів доступу к даним, зокрема додатки з низькою затримкою. Пошукові бази NoSQL призначені для аналітики частково структурованих даних. |
| Модель даних | Реляційна модель нормалізує дані та перетворює їх у таблиці, що складаються з рядків та стовпців. Схема чітко задає таблиці, рядки, стовпці, індекси, співвідношення між таблицями та елементами різними елементами бази даних. Така база даних забезпечує цілісність даних у співвідношеннях | Бази даних NoSQL надають різноманітні моделі даних, такі як «ключ-значення», документи, графи і колонки, оптимізовані для високої працездатності та масштабованості. |

| | **Реляційні бази даних** | **Бази даних NoSQL** |
|---|---|---|
| | між таблицями. | |
| Властивості ACID | Реляційні бази даних забезпечують наступний властивостей ACID, а саме: атомарність, безсуперечливість, ізольованість, надійність.<br>• Атомарність вимагає, щоб транзакція виконувалась повністю або зовсім не виконувалась.<br>• Безсуперечливість означає, що дані повинні відповідати схемі бази даних одночасно по транзакції<br>• Ізольованість вимагає, щоб паралельні транзакції виконувалися окремо одна від одної.<br>• Надійність передбачає здатність відновлення до останнього збереженого стану після непередбаченої помилки чи збою у системи або перебої у подачі живлення | Більшість баз даних NoSQL пропонують компроміс, пом'якшуючи жорсткі вимоги властивостей ACID заради більш набагато гнучкої моделі даних, котра дозволяє горизонтальне масштабування. Завдяки цьому, бали даних NoSQL – гарний вибір для прикладів використання з високою пропускною спроможністю та низькою затримкою, у яких є потреба в горизонтальному масштабуванні, без обмежень у рамках однієї інстанції. |

| | **Реляційні бази даних** | **Бази даних NoSQL** |
|---|---|---|
| Продуктивність | Продуктивність в основному залежить від дискової підсистеми. Для забезпечення максимальної продуктивності частіше потребується оптимізація запитів, індексів та структури таблиць. | Продуктивність зазвичай залежить від розміру кластера базового апаратного забесспечення, затримок в мережі та додатка, що звертається до бази даних. |
| Масштабування | Реляційні бази даних зазвичай масштабуються шляхом підвищення обчислювальних можливостей апаратного забезпечення або додавання окремих копій для робочих навантажень зчитування. | Бази даних NoSQL мають можливість розділу. Це досягається за допомогою шаблонам доступу с можливістю масштабування на основі розподіленої архітектури. Така можливість підвищує пропускну спроможність та забезпечує стійку продуктивність майже в безмежних масштабах. |
| API | Запити на зберігання та виведення даних робляться на мові SQL. Ці запити аналізує та виконує реляційна база даних. | Об’єктно-орієнтовні API дозволяють розробникам додатків без проблем зберігати та виводити структури даних. Завдяки використанню ключ |

| | Реляційні бази даних | Бази даних NoSQL |
|---|---|---|
| | | секцій, додатки можуть вести пошук по парам «ключ-значення», набором стовбців або частково структурованим документам, які містять серійні об’єкти та атрибути додатків. |

## 1.3 Постановка завдання та мети дослідження

Системи виявлення шахрайства базуються на концепції Big Data. Ці системи здатні використовувати різні джерела для виявлення зловмисницької діяльності, однак оцінка таких комплексів потребує розробки загального методу.

Мета цього дослідження полягає у підвищенні ефективності процесу виявлення шахрайської діяльності за рахунок комбінації потоку деталізованих записів з комутаторів разом з стандартизованими форматами.

Для створення такого джерела даних та його оцінки, необхідно вирішити такі наукові завдання:

1. Дослідити методи порівняння даних з мережного зонду для виявлення негативного впливу на роботу мережних елементів.
2. Проаналізувати методи моніторингу даних віртуалізованого середовища з резервуванням.
3. Дослідити потік CDR даних з IMS комутаторів та розробити алгоритм взаємодії системи розрахунку з системою моніторингу.
4. Визначити складові архітектури системи аналітики великих даних у залежності від джерела інформації та розробити схему етапів виявлення шахрайства.

5. Розрахувати показники оцінки середньозваженого значення часу затримки для визначення ефективності розробленого інтерфейсу на тестовому середовищі, що імітує роботу інформаційної мережі.

Таким чином, щоб досягнути поставленої мети необхідно виконати такі кроки:

1. Розробити схему виявлення шахрайства з поетапним розбиттям для оцінки роботи системи з урахуванням типу даних для обробки.

2. Удосконалити модель середовища обробки даних за допомогою записів з комутаторів базової мережі, а саме розробити алгоритм взаємодії системи розрахунку з системою моніторингу.

3. Створення симуляційного середовища, що наближене до інфокомунікаційної мережі, для практичного визначення ефективності системи на основі комплексного використання деталізованих записів та методу оцінки системи.

Реалізація даних завдань дозволить розробити методику оцінки систем виявлення шахрайства та оцінити наскільки розроблений алгоритм може вплинути на процес розпізнання шахрайства на інфокомунікаційній мережі.

## 1.4 Висновки до розділу

Перший розділ дослідження присвячено аналізу складових та функціоналу, що надають, різні системи виявлення шахрайства та аналітики ефективності роботи мережі. У розділі вивчені та проаналізовані основні елементи, процеси у системах Big data. Дослідження ретельно висвітлює структуру системи на базі AWS та розглядає її ключову особливість – сервіси як послуга. Історично, більшість систем розгортаються на фізичних серверах, що не завжди оптимально використовують надані ресурси. AWS дозволяє ефективне керування використаними ресурсами з урахуванням бюджету та коштів необхідних для користування сервісом. Система використовує машинне навчання, що надає змогу у автоматизованому розпізнанню шахрайства без ручного створення алгоритмів виявлення, які можуть призвести до

людської похибки. Було встановлено, що основним недоліком системи є використання лише одного джерела даних.

У розділі представлені системі на базі Apache Hadoop та наведений опис основних елементів. Такі системи використовують попередньо встановлені алгоритми виявлення шахрайства, які базуються на використані правил, шаблонів поведінки, списків з можливими шахраями або звичайними абонентами, на основі місцезнаходження абонента та машинного навчання. Було встановлено, що дані алгоритми можливо корегувати в залежності від отриманих результатів та на основі інформації яка є у користувача в наявності. В такому випадку, у даних алгоритмів є окремі вимоги до типу інформації та методів її направлення до системи. Даний аспект потребує подальшого вивчення джерел інформації, методів отримання даних з мережі та типів шахрайства, що можуть бути реалізовані на інформаційній мережі.

У розділі висвітлені основні елементи Apache Hadoop, наведено їх опис та методи роботи. Hadoop за своєю суттю є дуже ефективною у плані використання ресурсів системою з алгоритмами резервування модулів керування та зберігання, що робить її досить відмовостійкою. За рахунок відкритого коду, система дозволяє інтеграцію з корпоративними стандартами, такими як Amazon S3 замість вбудованого HDFS або навіть розробку власних фреймворків при необхідності, що є ще одною гарною особливістю. Основний недолік, що при виявленні помилки в коді доведеться його постійно при оновленнях переписувати або чекати виправлення від розробників, що не завжди має терміни виконання. Була наведена порівняльна характеристика між SQL та NoSQL базами даних, історична передумова для появи NoSQL. Основною відмінністю двох типів баз є те, що перший тип ефективно працює з структурованими типами даних та дуже жорсткими вимогами у роботі, доки другий тип працює з неструктурованим типом даних та має більш гнучкі вимоги у роботі з додатком.

# 2. ОСНОВНІ ВИДИ ШАХРАЙСТВА НА ІНФОКОМУНІКАЦІЙНІЙ МЕРЕЖІ ТА ТИПИ ДАНИХ ДЛЯ ЇХ ВИЯВЛЕННЯ

## 2.1 Шахрайство на інфокомунікаційній мережі

### 2.1.1 Шахрайство та його вплив на оператора інфокомунікаційної мережі

Шахрайство у інфокомунікаційних мережах поширюється з тревожним рівнем у всьому світі та є одним з найбільших джерел розмивання доходів для кожного оператора. По інформації CFCA (Communications Fraud Contro Association), глобальні втрати від шахрайства оцінювалися в 28,3 мільярда доларів США у 2019 році, що дорівнює 1,74% світового обсягу доходів інфокомунікацій у 2019 році, з урахуванням збитків від 5 найбільших видів шахрайства припадає 54% усіх втрат від шахрайств [26]. CFCA проводить опитування та дослідження кожні два роки. У 2023 році Communications Fraud Contro Association доповіла, що збитки від шахрайства зросли на 12% у порівнянні з 2021 роком, що дорівнює приблизно 38,95 мільярдам доларів США в 2023 році від 2,5% прибутку у сфері інфокомунікацій [66].

Таблиця 2.1 – топ методів та типів шахрайства в залежності від втрат.

| **Топ 10 методів шахрайства** | **Топ 10 типів шахрайства** |
|---|---|
| $1,92 млрд – шахрайство по підписці на додаток Subscription Fraud (Application) | $5,04 млрд – International Revenue Share Fraud (IRSF) |
| $1,82 млрд – шахрайство по сплаті (Payment Fraud) | $3,28 млрд – Arbitrage |
| $1,82 млрд – (PBX Hacking) | $2,71 млрд – Interconnect Bypass (e.g. SIM Box) |
| $1,82 млрд – IP PBX Hacking | $2,27 млрд – Domestic Premium Rate |

| **Топ 10 методів шахрайства** | **Топ 10 типів шахрайства** |
|---|---|
| | Service (In Country) |
| $1,82 млрд – Wangiri (Call Back Schemes) | $2,00 млрд – Traffic Pumping (includes: Domestic Revenue Share, TFTP) |
| $1,63 млрд – Зловживання мережею, елементами мережі або недоліки конфігурації мережі та мережевих елементів. | $1.76 млрд – комісійне шахрайство (Commissions Fraud) |
| $1.44 млрд – Dealer Fraud | $1,76 млрд – Перепродаж пристроїв або апаратного забеспечення |
| $1.34 млрд – Subscriber Fraud (Identity) | $1,49 млрд – Theft / Stolen Goods |
| $1.25 млрд – викрадення акаунту (Account Take Over) | $1,17 млрд – Friendly Fraud |
| $1.15 млрд – Внутрішне шахрайство аьо крадіжки співробітників (Internal Fraud / Employee Theft) | $0,98 млрд – Wholesale SIP Trunking Fraud |

У 2020 році AT&T з 171,8 млрд доларів річного доходу могла втратити 3.1 млрд доларів завдяки шахрайству, Telefonica Group з 49,2 млрд доларів річного доходу могла втратити 9,15 млн доларів, а Vodafone Group з 50,2 млрд доларів – 930 млн доларів.

Настороженність у цьому факті полягає в тому, що дві третини всіх втрат від шахрайства, особливо втрат від голосового шахрайства, пов’язані з міжнародним трафіком. Наприклад, у 2019 році збитки від International Revenue Share Fraud

(IRSF) перевищили 5 мільярдів доларів США, тоді як атаки Wangiri обійшлися мобільним операторам у 1,8 мільярда доларів США.

GSMA визначає інфокомунікаційне шахрайство як те, що вчиняється, коли процес, контроль або технічні недоліки навмисно використовуються, що призводить до фінансових чи інших втрат. GSMA визнає, що термін «шахрайство» визначено в багатьох національних правових системах, і оператори можуть використовувати різні визначення у своїх бізнесах і країнах. Зловмисники можуть або викрасти інфокомунікаційні послуги, або зловживати ними, щоб завдати збитків або обманом змусити невинних абонентів отримати величезні рахунки або викрасти особисті дані.

У світі високо взаємопов'язаних мереж 4G і 5G шахрайство може відбуватися з будь-якого місця. Не проходить і дня, щоб злочинці не намагалися шахраювати десь у світі, тому оператори повинні бути пильними щодо трафіку з будь-якого місця та до нього.

За даними CFCA, що були наведені у розділі вище, у 2019 році сума шахрайства в сфері інфокомунікацій склала 28,3 мільярда доларів США. 89% опитаних операторів сказали, що збитки від шахрайства в їхніх компаніях зросли або залишилися незмінними, однак зараз багато компаній повідомляють правоохоронним органам про набагато менше випадків.

Європейський оператор першого рівня (Tier-1) став жертвою широко поширеного PBX hacking fraud. Мережа першого рівня – це мережа, що може досягати кожну іншу мережу через  зєднання, за які вони не вносять плату (пірінг)[104, 105]. Після інциденту було зроблено аналіз який показав, що всього за три місяці на оператора було здійснено понад 200 спроб атак, що коштувало приблизно 130 000 євро прибутку.

Як зазвичай шахрайство впливає на мобільного оператора, окрім втрати доходу? Це шкодить бренду та репутації, збільшує витрати на обслуговування клієнтів, вимагає часу та робочої сили для усунення пошкоджень.

Як шахрайство впливає на споживача? Це спричиняє втрату грошей, довіри, особистих даних та тягне за собою наслідки, що вимагають часу на відновлення даних через неточні виставлення рахунків.

Можливо виділити декілька причин виникнення шахрайства на інфокомунікаційній мережі. Застарілі системи, які лежать в основі телефонної мережі, не були розроблені з урахуванням безпеки. Це не було проблемою, коли інфокомунікаційні мережі були закритим і контрольованим середовищем, де довіряли всім суб'єктам (операторам-монополістам). Але у наш час це спричинило появу різних вразливостей у мережі. На жаль, модернізація застарілих систем у глобальному масштабі неможлива у найближчи часи через високу вартість.

Інфокомунікаційні мережі складаються з різних, взаємопов'язаних технологій, послуг і продуктів, які зазвичай невідомі та не до кінця зрозумілі [65]. Це перетворює мережу на велику поверхню для атак. Усі учасники екосистеми мають адаптуватися до нових технологій, залишаючись пильними щодо можливих атак.

Оскільки ринок став більш лібералізованим, велика кількість різноманітних операторів залучилася до ринку. Як наслідок, неможливо переконатися, що всі сторони мають добрі наміри. Також не можна регулювати кількість операторів, а саме зменшувити їх кількість, оскільки це може зашкодити конкуренції та лібералізації, а також завадить розвитку нових технологій та різноманітності послуг.

Так чи інакше, вразливість мережі це лише наслідки причин, які можно вирішити або помякшити. Вразливості можливо класифікувати по 4 категоріям повязаними з протоколами та мережею, регулюванням, виставленням рахунків та людським фактором.

Інфокомунікаційні мережі — це взаємозв'язок між мережами телефонної мережі загального користування (ТМЗК), стільниковими та IP-мережами, усі з яких мають різні слабкі місця та вразливості. Зокрема, відсутність стандартів безпеки в сигналізації SS7 призводить до багатьох проблем, так як сама SS7 не

має жодних механізмів шифрування чи аутентифікації. Таким чином, оператори, що використовують SS7 (або будь-хто, хто має доступ до сигнальних каналів), можуть дивитися та модіфікувати повідомлення SS7 або взаємодіяти з системами SS7. Набір протоколів SIGTRAN був представлений як транспортний рівень для обміну повідомленнями SS7 через IP, який може використовувати TLS або IPSec. Однак немає наскрізної безпеки, і кожен транзитний оператор може змінювати складову повідомлення SS7.

З дерегуляцією та конвергенцією з Інтернет доступом, стало дуже просто отримати доступ до мереж SS7, тобто доступ більше не обмежується невеликою кількістю довірених операторів. Сьогодні оператори використовують механізми перевірки трафіку та правила фільтрації, щоб відкидати небажаний трафік[82].

Дійсно, зовнішнім організаціям стало простіше отримати частковий або повний доступ до SS7 через фемтостільники, канали SIP/PRI, домовленності з операторами інфокомунікацій (наприклад, додаткові послуги) або шляхом атаки на інфокомунікаційне обладнання. Шлюзи легального перехоплення, які операторам часто доводиться встановлювати для дотримання законодавства, також мають прямий доступ до SS7 і є джерелами вразливостей.

Протокол SS7 також не має стандартів відстеження маршруту виклику. Кожен комутатор має власну таблицю маршрутизації та вибір відповідного вихідного з’єднання на основі призначенного телефоного номеру, ціноутворення та комерційних угод. Таким чином, вони мають лише частковий перегляд маршруту виклику, що призводить до відсутності прозорості маршруту. З’єднання VoIP ще більше ускладнює відстеження дзвінків. Подібним чином під час сеансу зв’язку інформація про ідентифікацію абонента передається між операторами через систему сигналізації, яка надає інфокомунікаційні послуги. Однак цій інформації не можна довіряти як SS7 або більшість протоколів сигналізації на основі IP, бо вони не мають аутентифікації ідентифікатора абонента.

Бездротові та VoIP-мережі також часто не мають належної аутентифікації або шифрування, наприклад, між мобільним пристроєм і базовою станцією, що

призводить до можливості використання перехоплювача IMSI (IMSI Catchers) [85,90]. Більшість проблем у протоколах мобільних мереж вирішуються, починаючи з мереж третього покоління. Проте застарілі технології все ще широко використовуються, що відкриває можливість атак. Крім того, стільникові мережі та мережі VoIP успадковують деякі вразливості від ТМЗК, оскільки дзвінки все ще проходять через мережі ТМЗК [94]. LTE мережі також мають проблеми, пов'язані як з VoIP, так і зі стільниковими мережами, і можуть бути вразливими до атак з використанням різниці ціноутворення, DoS-атак і підміні ідентифікатора абонента (caller ID spoofing) [84].

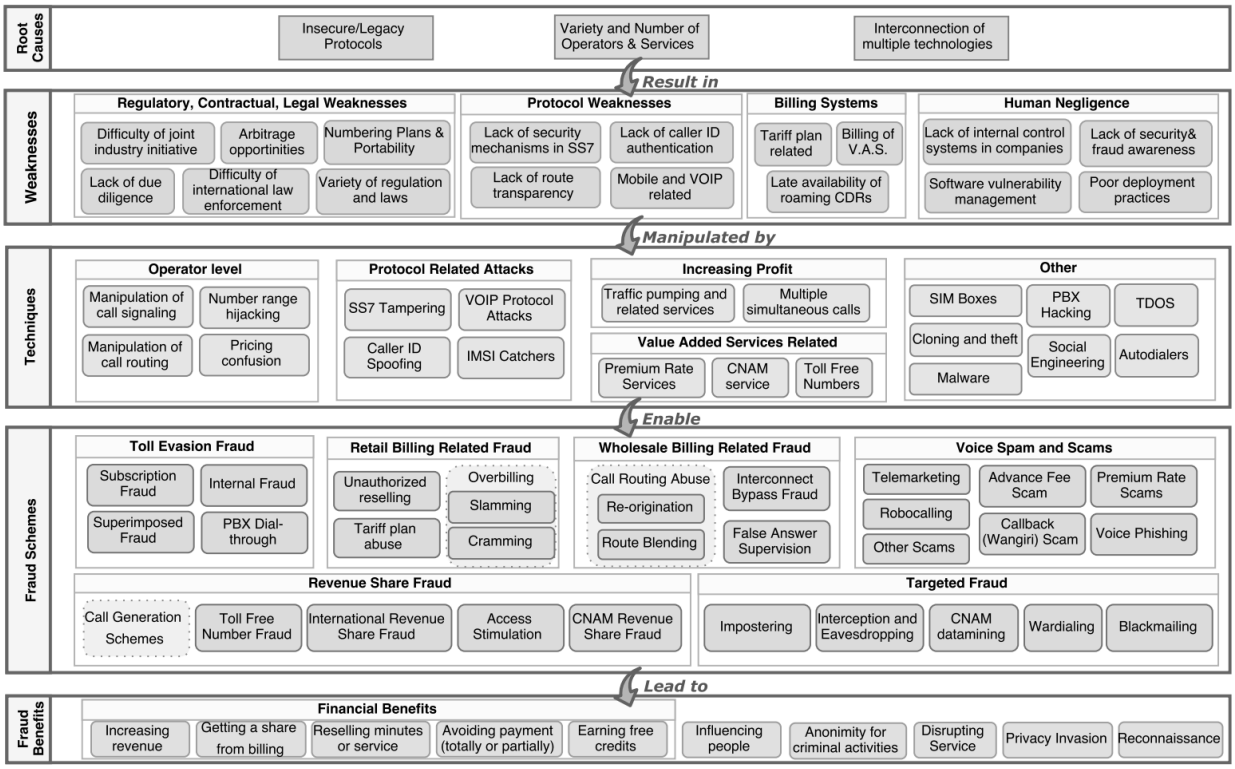



Рисунок 2.1 – Повна схема голосового шахрайства

Також, вистачає проблем у плані регулювання номерного ресурсу та взаємодію між оператором та країною на законодавчому рівні.

Плани нумерації дозволяють розкодувати номери телефонів і знайти оператора або тип послуги для даного номера. Стандарт E.164 описує структуру глобальної маршрутизації телефоних номерів з та призначає діапазони номерів

країнам (коди країн) [99]. Кожна країна має власний орган регуляції призначення та контролю національного діапазону номерів, але послуга перенесення номера розмиває межі. Не існує глобального плану нумерації, у якому було перелічено усі допустимі діапазони номерів у користуванні , хоча деякі бази даних дозволяють частковий пошук. Тому оператор не може знати напевно, чи використовується в даний момент телефонний номер в іншій країні. Протоколи VoIP використовують поняття контактів замість телефонних номерів. Однак, якщо виклик проходить через шлюз VoIP або ТМЗК, номер телефону має бути пов’язаний із контактом. Багато провайдерів OTT використовують номери телефонів для ідентифікації та аутентифікації своїх користувачів (наприклад, Viber, WhatsApp).

Екосистема телефонії втілює в собі велику різноманітність правил і законів. Поняття законності може значно відрізнятися залежно від країни та середовища передачі даних. Деякі країни забороняють використання VoIP, щоб захистити свої доходи за з’єднання міжнародних дзвінків. Деякі країни намагаються зв’язати провайдерів OTT тими ж правилами, що й операторів мобільного зв’язку. Загалом, потреба в регулюванні може бути не усвідомлена до того, як системою почнуть маніпулювати. Таким чином, регуляторам може бути важко передбачити потреби регулювання.

Відсутність співпраці є ще одним недоліком екосистеми телефонії. Правоохоронні органи мають труднощі з дотриманням міжнародного права, що ускладнює ідентифікацію шахраїв, навіть якщо шахрайство виявлено. Крім того, незважаючи на наявність міжнародних організацій, не вистачає спільних галузевих ініціатив для боротьби з шахрайством. Завдяки проблемам конфіденційності та конкурентності, оператори зазвичай не бажають ділитися своїми умовами ціноутворення, варіантами маршрутизації чи інформацією стосовно шахрайства[]. Крім того, не всі оператори мають однакові стимули для боротьби з шахрайством. Дійсно, іноді збитки через шахрайство в одного оператора можуть принести вигоду іншому, невинному оператору на додаток до

шахрая. В інших випадках боротьба з дрібним шахрайством може бути дорожчою, ніж збитки від діяльності шахрайя.

Наявність великої кількості операторів призводить до неминучої необхідності партнерства між ними. Відсутність належної обачності в цих партнерських угодах робить трафік дзвінків вразливим для шахрайства, якщо одна зі сторін має шахрайські наміри. Особливо, транзитні оператори конкуренти можуть ігнорувати якість маршрутів і використовувати дешеві маршрути для розвитку свого бізнесу.

З появою нових технологій і послуг складність механізмів розрахунку послуг зросла. Будь-які помилки в процесі розрахунку (наприклад, неточне або пізне виставлення рахунків, помилки у відстеженні передоплаченого кредиту) можуть маніпулювати шахраї. У більшості випадків оператори неохоче змінюють застарілі системи розрахунку послуг через високу вартість і проблеми із зворотною сумісністю. Тому, в складних тарифних планах також можна маніпулювати помилками.

Виставлення рахунків за додаткові послуги є ще одним слабким місцем, оскільки це додає сторонній функціонал до системи. Через високу плату вони можуть призвести до значних збитків. Операторам слід бути обережними у визначенні номерів з додаткових послуг та при реєстрацію суб'єктів, які використовують ці номери. Шахраї часто користуються складними мережами посередників і постачальників послуг, тому їх важко ідентифікувати.

Людський фактор теж грає свою роль як вразливість мережі. Завдяки наівності, необачливості, незнанням елементарних правил безпеки або недостатнім рівенем конфіденціальності та безпеки у компаніх шахраї можуть маніпалювати працівниками, щоб отримати необхідну інформацію або необхідний доступ.

### 2.1.2 Основні типи та методи реалізації шахрайства

Згідно зі звітом CSFA, найпоширенішими типами шахрайства з найбільшою часткою втрати прибутку є International Revenue Share Fraud (IRSF), Arbitrage, Interconnect Bypass (наприклад, SIM Box), Domestic Premium Rate Service (у

країні), Traffic Pumping, що містить у собі Domestic Revenue Share, TFTP. Наведені типи шахрайства складаються з менших типів або комбінуються між собою. Тому окрім найпоширеніших видів, у підрозділі будут розглядатися та більш меньші[3].

Шахрайство **IRSF** використовує преміальні мобільні тарифи, які потім абоненти набирають мимоволі. Цей вид можна реалізувати декількома способами:

- Шахраї оформлюють підписку на оренду преміального телефонного номеру.
- Шахрай взламує телефонні системи підприємства (взламування PBX) та здійснює дзвінки на цей преміальний номер.
- Шахраї використовують взламані телефони або вкрадені пристрої/SIM-карти.

Дзвінки, частіше за все, відбуваються в неробочий час, і компанії розуміють, що вони були зроблені лише тоді, коли приходить час сплачувати рахунок. Багато зловмисників обирають повільні та непомітні атаки, щоб обійти правила ідентифікації атак. Підприємствам також не вистачає видимості подальшої діяльності, що не дозволяє їм виявити та усунути IRSF на ранніх стадіях. Давайте розглянемо два приклади такого шахрайства IRSF на базі OTP та Wangiri[30].

Етапи IRSF на основі OTP (рис 2.1):

1. Зловмисник ініціює атаку вручну або за сценарієм на веб-сторінку, щоб активувати одноразові паролі на основі голосових або SMS-повідомлень. Залежно від результатів зловмисники можуть вибрати атаки великого обсягу або атаки з низьким і повільним впливом.
2. Підприємство, орієнтоване на споживача, пересилає запит OTP до постачальника хмарного зв’язку.
3. Провайдер пересилає запит оператору. Цей запит необхідний, оскільки в регіоні задіяно кілька мережевих провайдерів, перш ніж OTP зможе досягти цільового споживача.
4. Скомпрометований оператор зв’язку, вступивши в змову зі зловмисником, пересилає запит на номер IPR, а не на одержувача.

5. «Завершити» дзвінок на номері IPR дорого; і врешті-решт саме бізнес, орієнтований на споживача, повинен покрити збитки.
6. Зловмисники отримують величезні прибутки від своєї частки шахрайства, зазвичай 1 долар або більше за транзакцію.

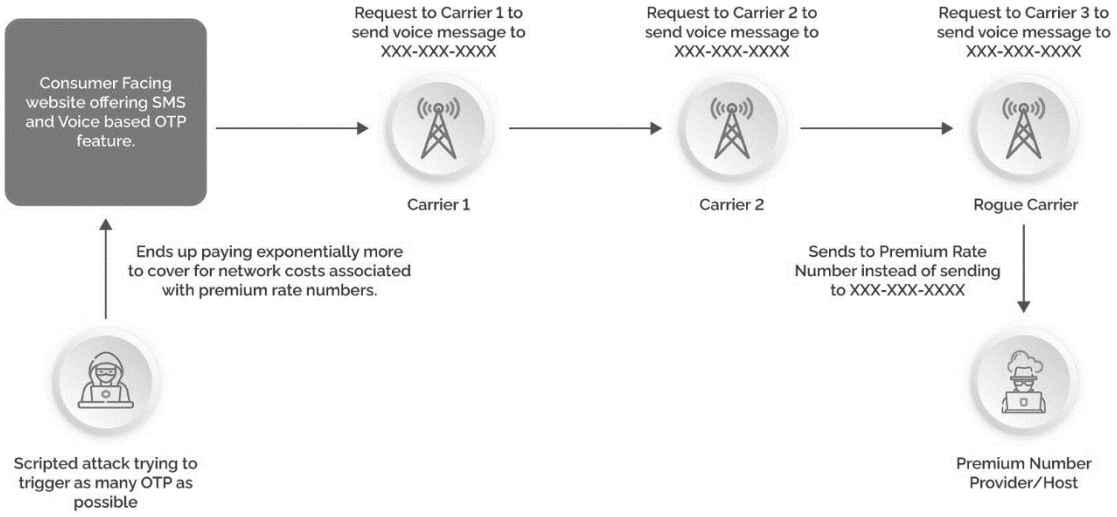



Рисунок 2.2 – Схема IRSF шахрайства на базі OTP

**Wangiri** — японське слово, що означає «одне (кільце) і вирізати». Це телефонне шахрайство, коли зловмисники за допомогою хитрості змушують вас дзвонити на платні номери (рис 2.3). Шахрай налаштує систему (наприклад, за допомогою ботнетів) для того, щоб телефонувати великої кількості випадкових номерів. По кожному дзвінку робиться лише одна спроба, а потім кладеться слухавка, залишаючи пропущений виклик на телефоні одержувачів. Користувачі часто бачать пропущений дзвінок і, вважаючи, що це був справжній дзвінок, передзвонюють на пропущений номер. Також існує SMS варіант такого шахрайства, коли шахраї надсилають повідомлення, пропонуючи клієнтам передзвонити на певний номер або навіть надіслати SMS. Типовими тривожними ознаками для такого роду інфокомунікаційного шахрайства є стрибки трафіку на пункти призначення з високою ціною тарифікації, які інфокомунікаційні компанії повинні мати можливість відстежувати за допомогою своїх внутрішніх систем.

Ключовим тут є, що бізнес повинен стежити за тим, які номери автоматично набираються.

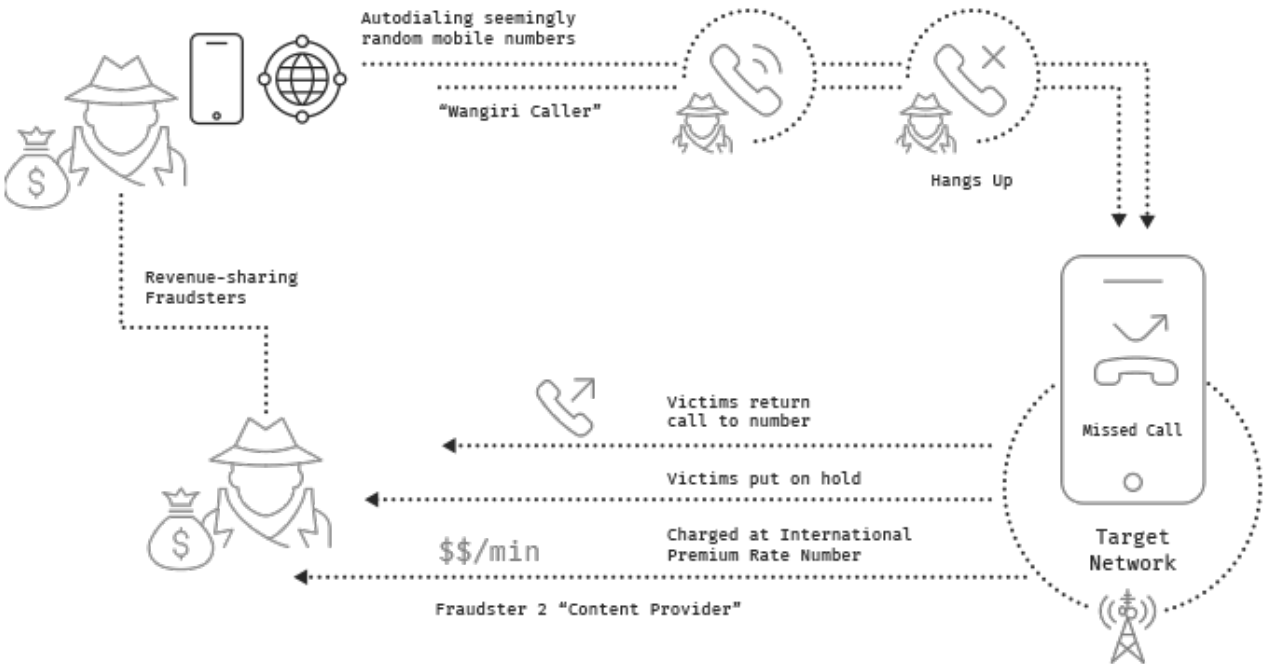



Рисунок 2.3 – Схема Wangiri шахрайства

**Interconnect bypass fraud**, також відоме як шахрайство з SIM-боксом, використовує так звану ціну з'єднання, щоб зробити телефонні дзвінки дешевшими [31]. Щоб зрозуміти це, розглянемо сценарій з двома операторами в різних країнах:

1. Клієнт оператора А дзвонить клієнту оператора Б.
2. Оператор А стягує зі свого клієнта плату за хвилину.
3. Оператор В стягує плату з оператора А за надання дзвінка його клієнту.

Остання плата, на якій виклик завершується, є ціною за з'єднання. Ці ціни дуже різняться залежно від контрактів між двома операторами. Деякі з них мають високу ціну, поки ціна інших наблизена до нуля. Ось тут і з'являється оператор-шахрай. Вони перенаправляють ці міжнародні дзвінки за допомогою SIM-боксу або шлюзу GSM, фактично захоплюючи з'єднання, щоб отримати дешевші тарифи за з'єднання між абонентами. По суті, дзвінки на великі дистанції

робляться набагато дешевшими, але абонент платить ту саму ціну, а шахрайська інфокомунікаційна компанія отримує прибуток від цієї різниці. Це також впливає на задоволеність клієнтів інфокомунікацій, оскільки найчастіше якість цих дзвінків буде нижчою від стандартних міжнародних дзвінків. На рисунку 2.4 ми можемо побачити приклад шахрайства з обходом VoIP, коли дзвінок маршрутизується через Інтернет замість узгоджених маршрутів оператора зв'язку.

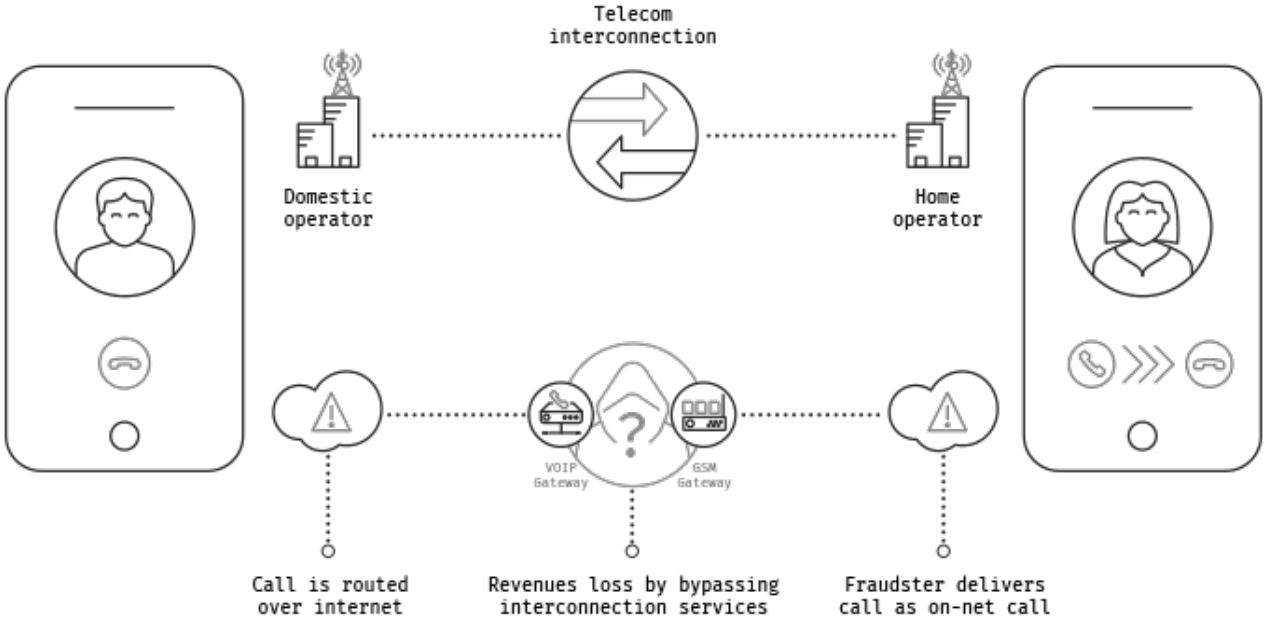



Рисунок 2.4 – Схема Interconnect bypass fraud шахрайства

**Arbitrage** — це загальна практика отримання прибутку від різниці в ціні. У світі інфокомунікацій ці відмінності проявляються в тарифікаціях на великі відстані між країнами. Подібно до міжнародного обхідного шахрайства, це може знизити міжнародні витрати для клієнтів, але також відкрити двері для шахрайських компаній, які непомітно стають між операторами. Вони стверджують, що з'єднуються безпосередньо з країни А в країну Б, тоді як насправді вони здійснюють дзвінок через країну з нижчим тарифом.

**Traffic pumping**, також відоме як стимулювання доступу, є сумнівною практикою, за допомогою якої деякі місцеві оператори телефонного зв'язку в

сільській місцевості Сполучених Штатів збільшують обсяг вхідних дзвінків до своїх мереж, щоб отримати прибуток від значно збільшених компенсаційних комісій між операторами. Щоб підштовхнути конкуренцію за діючих місцевих операторів зв'язку (ILEC), FCC дозволяє сільським конкурентним місцевим операторам зв'язку (CLEC) і діючим місцевим операторам зв'язку (ILEC) стягувати високу плату за кінцевий доступ за виконання дзвінків, які вони приймають від операторів міжстанційного зв'язку (IXCs).

**Смішинг/SMS-фішинг (Smishing/SMS Phishing)** — це практика масового надсилання SMS з метою отримання особистої інформації від особи, яка отримує повідомлення. Простої системи для моніторингу реєстрацій і транзакцій від служб B2B, має бути достатньо, щоб переконатися, що оператор інфокомунікацій не допомагає смішинг бізнесу [7].

**SMS взламування (SMS Bypass)** - зловмисники можуть зламати SMS-центр мобільного оператора або навіть контролювати його на рівні сигналізації, щоб використовувати для розсилки трафіку по всьому світу. Цей трафік може спонукати споживачів здійснювати дзвінки на преміум номери або навіть містити віруси чи інше шкідливе програмне забезпечення, яке може заразити телефон одержувача.

**SMS Шахрайство на основі шкідливого програмного забезпечення**. Найпопулярнішою реалізацією такого шахрайства став FluBot. Вперше виявлений у грудні 2020 року, FluBot набув популярності у 2021 році та скомпрометував величезну кількість пристроїв у всьому світі, у тому числі значні випадки в Іспанії та Фінляндії. Зловмисне програмне забезпечення було встановлено за допомогою текстових повідомлень, які просили користувачів Android натиснути посилання та встановити додаток для відстеження доставки посилки або прослухати фальшиве повідомлення голосової пошти. Після встановлення шкідлива програма, якою насправді був FluBot, запитує Android дозволи. Потім хакери використовують ці дозволи, щоб викрасти облікові дані банківського додатка або деталі рахунку в криптовалюті та вимкнути вбудовані механізми безпеки. Цей тип зловмисного програмного забезпечення міг поширюватися як лісова пожежа через його

здатність отримати доступ до контактів інфікованого смартфону. Потім на ці номери надсилалися повідомлення з посиланнями на зловмисне програмне забезпечення FluBot, що сприяло подальшому його поширенню [7,33].

**SMS Bypass** — це практика, коли несумлінні SMS агрегатори використовують несанкціоновані або навіть незаконні маршрути для доставки SMS-повідомлень за найнижчою ціною. Така практика шкодить операторам, позбавляючи їх законних доходів за доставку. Щоб зрозуміти це, розглянемо приклад з двома операторами в різних країнах:

1) Абонент оператора А надсилає SMS абоненту оператора Б.

2) Оператор А стягує зі свого абонента плату.

3) Оператор В стягує з оператора А плату за надання SMS своєму абоненту.

Остання плата, на якій SMS доставляється отримувачу, є ціна за доставку. Ці ціни дуже різняться залежно від домовленностей між двома операторами. Деякі з них мають високу ціну, поки ціна інших наблизена до нуля. Ось тут оператор-шахрай і починає діяти. Вони перенаправляють ці міжнародні SMS-повідомлення за допомогою SIM-боксу або GSM-шлюзу, фактично захоплюючи з'єднання, щоб отримати дешевші тарифи за доставку повідомлення до абонента. По суті, SMS-повідомлення на великі дистанції робляться набагато дешевшими, але абонент платить ту саму ціну, а шахрайська інфокомунікаційна компанія отримує прибуток від цієї різниці.

Поширеним симптомом SMS-шахрайства є аномальні стрибки обсягу SMS-трафіку від одного оператора до іншого. Нещодавно австралійський оператор раптово почав отримувати величезні обсяги трафіку SMS з африканської країни. Щоденні обсяги SMS, які зазвичай сягали сотні, досягали 145 000 повідомлень на день, що коштувало австралійському оператору 10 000 євро на день у вигляді комісії за доставлення SMS, на яку не можна було виставляти рахунок. Якби випадок залишився непоміченим, австралійський оператор втратив би 300 000 євро через SMS-шахрайство лише за місяць [25].

**Викрадення діапазону номерів (Number range hijacking)** відбувається, коли оператор з шахрайськими намірами пропонує низькі ціни на діапазон

номерів призначення чим приваблює трафік від інших опепаторів. Наприклад, на рисунку 2.5 є декілька маршрутів до цільового оператора. Нехай, маршрут 1 та 3 є найбільш звичайними маршрутами. У цьому випадку оператор 3 вирішив зробити рекламу на дуже дешевому тарифі (для дуже маленького діапазону номерів), тому вихідний оператор може вибрати маршрут 2 для доставки дзвінків. У цьому випадку діапазони номерів жертви можуть бути вкрадені та перенаправлені через шахрайське обладнання. Ця техніка використовує необачність у домовленності між операторами зв'язку [65].

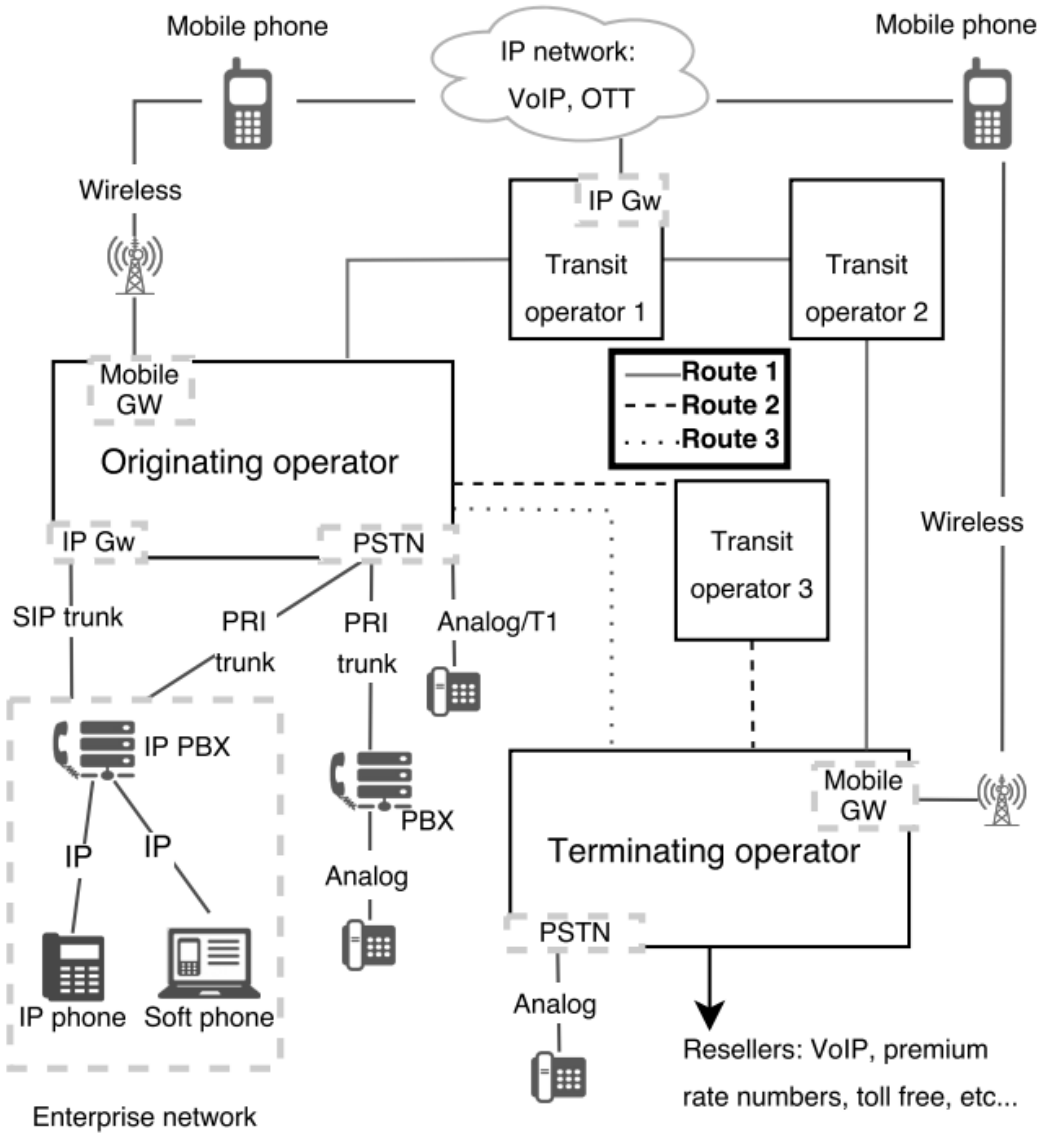



Рисунок 2.5 – Схема Number range hijacking шахрайства

**Викрадення BGP (BGP hijack)** має схожу техніку як і Number range hijacking. В обох випадках частина трафіку перенаправляється зловмисним субєктом, який пропонує неправдиву (або оманливу) інформацію. У випадку це реклама префіксів [81].

**Маніпулювання маршрутизацією викликів (Manipulation of call routing)** можливе при наявності у оператора над транзитом через їх мережу (легитимно або через викрадення). Транзитний оператор з шахрайськими намірами може переадресовувати дзвінок або перенаправити його нелегітимними маршрутами для виконання різних шахрайських схем. У разі короткої зупинки виклику транзитний оператор безпосередньо припиняє виклик (наприклад, до IVR) замість того, щоб пернаправити виклик до пункту призначення. Він також може вибірково зупиняти лише деякі дзвінки. Через відсутність прозорості маршруту вихідний оператор не може знати, чи дзвінок було направлений по звиайному маршруту та чи дійшов він правильного пункту призначення.

**Маніпулювання сигнальними повідомленнями виклику (Manipulation of call signaling)** також легко реалізується оператором. Наприклад, ідентифікатор вихідного абонента можна змінити, щоб підробити джерело виклику, яке може вплинути на розрахунок дзвінку. Сигнали встановлення виклику можуть бути модифіковані, щоб відповідь на виклик почався до того, як на нього фактично відповість абонент (рання відповідь) або щоб не роз'єднати виклик негайно (пізнє роз'єднання) [97]. Дзвінок буде довшим, ніж слід, що вплине на ціну такого виклику. До данного типу шахрайства можно віднести **False Answer Supervision (FAS)**. У FAS транзитні оператори використовуя шахрайскі методи можуть збільшити свій дохід з кожного сеансу звязку, виконуючи одну з наступних дій:

- Помилкова відповідь (False answer): (також називається шахрайством з короткою зупинкою) оператор перенаправляє виклик (коротко зупиняє його) на записане повідомлення та починає стягувати плату, замість того, щоб перенаправити виклик до цільової мережу.

- Рання відповідь (Early answer): оператор збільшує тривалість дзвінка шахрайським шляхом, наприклад, відповідаючи на дзвінок і відтворюючи фальшиву мелодію дзвінка, доки абонент дійсно не відповість.
- Пізнє роз'єднання (Late disconnect): оператор затримує передачу повідомлення про роз'єднання виклику стороні, що телефонує, і тому виставляє рахунок за більш тривалу розмову.

Згідно з опитуванням, проведеним у 2013 році, FAS був найпопулярнішим шахрайством серед великих операторів зв’язку, що надають свої мережеві ресурси. Це також може завдати шкоди репутації звичайних операторів, оскільки вони можуть отримувати скарги від клієнтів щодо неправильного розрахунку за послуги зв’язку.

**Плутанина в ціноутворенні** — це використання кількох і різноманітних у ціні тарифних планів, щоб заплутати абонентів щодо реальної ціни послуги. Такі оператори постійно пропонують нові пропозиції та спеціальні вступні знижки, щоб бути конкурентоспроможними [65], але швидко змінюють ціни після реєстрації нового абоненту.

**Телефонні номери по високим цінам (Premium Rate Numbers PRN)** використовуються для надання широкого діапазону послуг як казино, живі чати, послуги для дорослих через голосовий дзвінок або SMS. Для покриття наданих послуг, ціна на дзвінок PRN набагато вища ніж за звичайні. У більшості країн фіксований діапазон номерів використовується для PRN, що дозволяє абонентам легко їх розрізняти, але не всюди. Користувачі іноді плутають звичайний номер з PRN та випадково роблять виклик. Такі номери можуть використовуватися у шахрайстві, коли послуга не була надана, ціна сервісу не була прозорою або трафік був штучно перенаправлений на ці номери. Сервіси, що зловживають PRN, маніпулюють між торговими посередниками та номерними планами у яких діапазон PRN не можуть бути чітко виявлені абонентами. Абагто веб сайтів надають послуги PRN, які дають кешбек у увипадку, якщо дзвінки досягать цього преміального номеру.

**Служба пошуку CNAM (Caller Name)** надає рядок імені власника номеру телефона з 15 символів, щоб допомогти користувачам легко ідентифікувати абонента. У США оператори відповідають за пошук CNAM для дзвінків, отриманих їхніми клієнтами. Послуга CNAM зазвичай постачається як частина пакета стаціонарного зв'язку. Нажаль, у Північній Америці централізованої бази даних CNAM немає. Натомість кілька незалежних постачальників CNAM дозволяють операторам шукати інформацію CNAM за плату [89]. Шахраї можуть використовувати службу CNAM, щоб зареєструвати фальшиве ім'я власника для свого номера телефону або реалізовувати шхрайство на основі ціноутворення.

**Безкоштовні номери (Toll free numbers)** — це телефонні номери, за які абонент не стягує жодної плати. Натомість дзвінок оплачується безкоштовним клієнтом (отримувачу дзвінка), яким зазвичай є послугами кол-центру. Для безкоштовних номерів використовується префікс, призначений регулятором. Для безкоштовних номерів стягнення плати зворотне: безкоштовний клієнт платить постачальнику безкоштовних номерів (зазвичай кінцевому оператору) за всі вхідні дзвінки. Безкоштовні провайдери зберігають частину прибутку та передають її оператору, який телефонує, оскільки абонент не платить за дзвінок [99].

**Втручання в SS7** стороннім суб'єктам стало можливим завдяки легкому доступу до мереж SS7. Це дозволяє робити атаки для визначення місцезнаходження пристроїв користувача, перехоплення дзвінків або відмова в обслуговуванні мережею.

**Атаки на протокол VoIP** можуть використовувати недоліками реалізації, мережеве обладнання на якому працює сервіс мережевою платформою або рівнем програмного забеспечення. Зазвичай до таких атак належать сканування SIP, підміна реєстрації, перенаправлюючі атаки, розрив сеансу зв'язку, перезавантаження SIP телефонії та вставка аудіо. Системами розрахунку вартості також можлиов маніпулювати через VoIP атаки [65].

**Перехоплювачі IMSI (IMSI catchers або stingray)** [85, 90] — це несправжні базові станції GSM, які використовуються для ідентифікації телефонів навколо

них (перехоплення IMSI абонентів), перехоплення дзвінків і комунікацій або навіть для розсилання спаму та шахрайських повідомлень. Перехоплювачі IMSI маніпулюють відсутністю автентифікації від мережі до пристрою в GSM. Таку базову станцію можна побудувати за допомогою обладнання операторського рівня або програмного забезпечення з відкритим кодом та дешевого апаратного забеспечення. Абонентський пристрій обманом намагається підключитися до фальшивої базової станції, і зазвичай мобільний пристрій змушений не використовувати шифрування або перейти до незахищеного режиму (наприклад, з 3G до 2G). Новітніші мобільні протоколи (наприклад, LTE) використовують аутентифікацію, але не захищені від таких атак по двум причинам:

- по-перше, міг статися витік (або конфіскація) ключів аутентифікації;
- по-друге, перехоплювачі IMSI можуть зловживати вразливими місцями в стеках протоколів [].

Для виявлення перехоплювачів IMSI можуть використовувати перевірку розбіжностей у прийнятих конфігураціях мережі.

**Підробка ідентифікатора абонента (Caller ID spoofing)** передбачає передачу фальшивих ідентифікаторів абонента у систему сигналізації. Незважаючи на те, що існують певні законні способи використання спуфінгу ідентифікатора абонента [100], це безумовно є каталізатором голосового шахрайства. Транки PRI і SIP дозволяють передавати фальшивий ідентифікатор абонента в мережу SS7 в результаті відсутності аутентифікації ідентифікатора абонента. Різноманітні онлайн-сервіси та мобільні додатки забезпечують підробку ідентифікатора абонента через підключення шлюзу IP/ТМЗК постачальника послуг. Підробити ідентифікатор абонента між двома програмами VoIP ще простіше, оскільки ідентифікатор абонента можна вставити в SIP-запити.

Окрім використання данного типу у шахрайстві, підробка ідентифікатора абонента може нашкодити різним сервісам (таким як банківські системи, голосова пошта або сервіси невідкладної допомоги), де інформація ідентифікатора може використовуватися для аутентифікації або визначення місцехнаходження користувачів.

Робоча група STIR (Secure Telephony Identity Revisited) намагалася надати механізм аутентифікації заголовоків SIP, щоб аутентифікувати ідентифікатор абонента. Нажаль, операторати інфокомунікацій неохоче розгортають таке рішення через накладні витрати. Там паче, за допомгою хмарних VoIP сервісів, телефонні номери стали дуже дешевими та шахраї можуть легко отримати пачку телефонних номерів та змінювати аутентифіковані номери бистріше, ніж їх зможуть заблокувати.

Шахраї можуть зловживати пристроями користувача та SIM-картами за допомогою **крадіжки та клонування**. У мережах CDMA клонування телефону здійснюється шляхом перепрограмування електронного серійного номера пристрою та мобільного ідентифікаційного номера. У мережах GSM телефони ідентифікуються за номером IMEI. Фальсифікація IMEI може бути корисною в деяких країнах, щоб обійти державний контроль пристроїв або уникнути внесення вкрадених телефонів до чорного списку. Заміна SIM (SIM swap) – це послуга, яку інфокомунікаційні оператори надають абонентам для реєстрації існуючого номера телефону на новій SIM-картці. Цим сервісом можуть маніпулювати шахраї, щоб отримати право власності на телефонну лінію. З цією метою шахрай зв’язується з оператором, стверджуючи, наприклад, що SIM-карту вкрали, і використовує методи соціальної інженерії, щоб видати себе за власника SIM-карти. Якщо шахрай зможе переконати оператора зареєструвати нову SIM-карту на певний номер телефону, він зможе генерувати дзвінки, які будуть оплачуватися з чужого рахунку. Заміна SIM-карти також може впливати на двофакторні механізми автентифікації, включаючи банківські [65].

**SIM-бокси або шлюзи GSM** — це пристрої, які можуть діяти як шлюз до мобільної мережі (наприклад, GSM) і надавати з’єднання до VoIP або PRI. Ці пристрої використовуються для забезпечення мобільного підключення до приватних або корпоративних АТС (PBX). Пристрій, по суті, складається з одного або кількох мобільних модемів, до яких можна приєднати SIM-карти, модемами керує комп’ютер, який перетворює дзвінки на VoIP або ЦМзІП (Цифрова Мережа з Інтегрованими Послугами або ISDN). SIM-бокси мають законне використання

(наприклад, надання шлюзів GSM для корпоративних систем PBX), яке дозволено операторами та регуляторами. Однак існує багато шахрайств, які використовують SIM-бокси, зокрема interconnect bypass та IRSF [65, 96].

**Автоматичні системи набору номера (Autodialers)** — це системи, які автоматично набирають телефонні номери випадковим чином або за попередньо визначеним списком [98]. Після того, як інша сторона відповіла на дзвінок, комплексна система може проаналізувати вхідний аудіопотік, щоб переконатися, відповіла справжня людина чи автовідповідач. Вони можуть або поставити попередньо записане повідомлення, або підключити дзвінок до іншої людини.

**DoS телефонії (Telephony Denial of Service -TDoS)** здійснюються шляхом надсилання дуже великого обсягу трафіку викликів на цільовий номер, щоб зменшити системні ресурси (наприклад, пропускну здатність магістралі) і порушити телефонну службу цільового клієнта. Існують різні способи у реалізації TDoS-атаку, наприклад, зібрати людей у соціальних мережах, щоб зателефонувати на певний номер, або використовувати autodialers. Завдяки об'єднанню телефонії та Інтернету атаки TDoS стали легшими. Зловмисник може використовувати шлюз VoIP-ТМЗК для створення дешевих дзвінків, програмне забезпечення для генерації викликів і деяке записане аудіо повідомлення, щоб зупинити ціль за реалістичним сценарієм [21]. Шлюз VoIP-ТМЗК може бути безкоштовним програмним забезпеченням IP-АТС (наприклад, Asterisk) з доступом до SIP транку. Іншими методами може бути використання скомпрометованої приватної або корпоративної АТС, ботнету або онлайн-сервісу TDoS [65].

**Соціальна інженерія** – це процес маніпулювання людьми, щоб вони діяли певним чином або повідомили конфіденційну інформацію [65]. Атаки соціальної інженерії використовують, щоб скористатися людською необачністю про інформаційну безпеку вбо необізнаність про шахрайство. Деякі приклади атак соціальної інженерії: змусити співробітників компанії надати свої паролі або переконати їх зателефонувати на певний номер телефону. Телефонія була кращим каналом для соціальної інженерії через довіру людей до телефону та тому, що

легше видати себе за іншу особу по телефону. Зловмисне програмне забезпечення, що заражає смартфони та телефони VoIP, може викрадати особисті дані, наприклад, допомагаючи в соціальній інженерії, але також може ініціювати дзвінки або надсилати SMS повідомлення.

**Шахрайство з підпискою (Subscription fraud)** — це використання викрадених ідентифікаційних даних або надання фальшивої інформації під час підписки на послугу, щоб уникнути плати за обслуговування. Наприклад, підписка на SIM-картку для облікового запису з передплатою за допомогою неправдивої інформації. Для виявлення шахрайства данного типу можливо використати аналіз даних та інші методи класифікації [65].

**Внутрішнє шахрайство (Internal fraud)** зазвичай здійснюється співробітниками інфокомунікаційної компанії, які мають доступ до облікових даних користувачів, тарифних планів та системи виставлення рахунків. Співробітники-шахраї можуть, наприклад, деактивувати обробку рахунків для певних облікових записів, підробити записи дзвінків або маніпулювати тарифними планами, щоб уникнути або зменшити плату за надані послуги [65, 82].

**Wardialing** — це використання автоматичних систем набору номера для сканування діапазону телефонних номерів, наприклад, для ідентифікації модемів, факсів або облікових записів голосової пошти. Використовується для розвідки у цільовій компанії та подальшої атаки. У CNAM data mining шахрай дзвонить собі кілька разів, підробляючи ідентифікатор абонента інших номерів, щоб отримати інформацію про ім’я абонента для цих номерів. Зазвичай,  на такі дзвінки не відповідають, але у бьдь-якому випадку оператор, на якого підписан зловмисник, змушений шукати підроблений номер у CNAM.

**CNAM revenue share** є не дуже відомим механізмом шахрайства. Ймовірно, ця схема стала можливою через незрозумілий і дерегульований характер служби CNAM. Існує багато постачальників послуг CNAM (Caller NAME). Оператори покладаються на них, щоб надати своїм клієнтам інформацію про ім’я абонента. Коли цільова телефонна компанія отримує дзвінок, вона за окрему плату виконує

запит (плата за запит CNAM). Ця компенсація відбувається для кожного дзвінка, де ім’я абонента відображається для викликаної сторони, навіть якщо на дзвінок немає відповіді.

CNAM revenue share fraud схоже на інші схеми шахрайства з розподілом доходу, де шахрай генерує трафік дзвінків до партнера постачальника послуг, який ділиться частиною свого доходу з шахраєм. Проте шахрай повинен «ініціювати» дзвінки з номерів частки доходу (замість «припиняти»). Це пов’язано з тим, що служба пошуку CNAM використовує знайдений ідентифікатор абонента для визначення партнера, який розподіляє дохід. Крім того, генерування дзвінків обходиться шахраям дешево, оскільки на дзвінки зазвичай не відповідають.

З іншого боку, ціна за запит CNAM зазвичай дуже мала, і щоб отримати значний прибуток, шахрай повинен генерувати велику кількість дзвінків з номера, що використовується в шахрайстві. Оскільки генерувати тисячі дзвінків з одного телефону непрактично, шахрай зазвичай підробляє свій ідентифікатор абонента та використовує кілька автоматичних систем набору номера для генерації дзвінків (рис. 2.6).

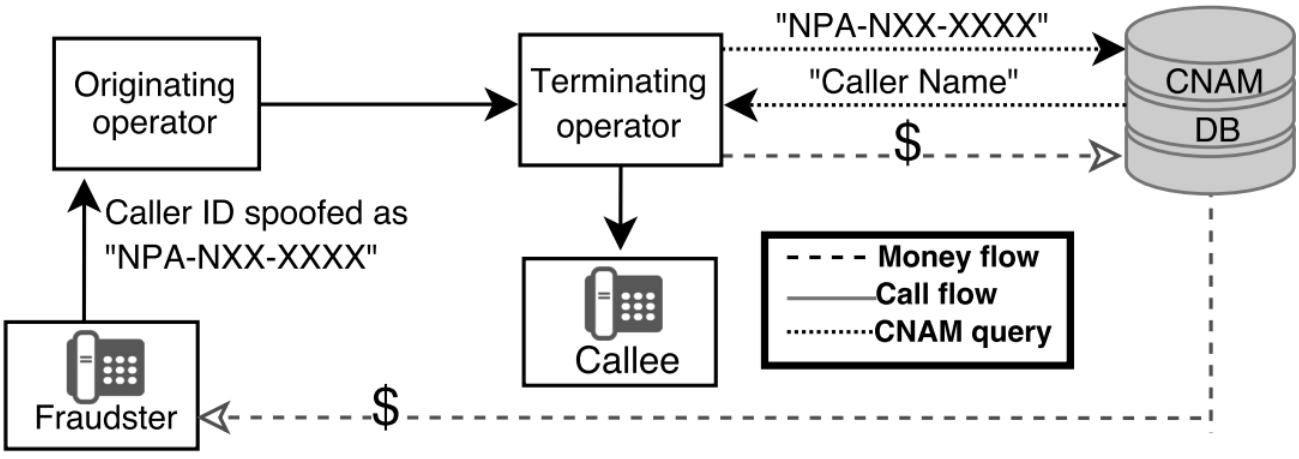



Рисунок 2.6 – Схема шахрайства CNAM revenue share

Для абонентів інфокомунікаційних послуг помітним ефектом цього шахрайства буде велика кількість пропущених дзвінків на стаціонарні телефони. Ці дзвінки, швидше за все, сприйматимуться клієнтами як голосовий спам (або

виклики ping). Насправді служби CNAM також часто використовуються телемаркетингами, щоб отримати додатковий прибуток від своїх дзвінків, що може пояснити, чому телемаркетери рідко випадково підробляють свої ідентифікатори абонентів.

Реклама CNAM шахрайства в Індії. Провайдер CNAM пропонує кол-центру в Індії розподіл прибутку. У цьому прикладі 34 мільйони дзвінків було згенеровано протягом 10 місяців, причому 39% дзвінків призвели до запиту CNAM. Середня частка доходу становила 1,15 доларів США за тисячу пошуків (запитів) CNAM, що дало загальний дохід 15 372 дол. Хоча важко повністю довіряти цій рекламі, вона все ж дає цікаве уявлення про масштаб цього шахрайства. По даним іншої реклами дохід за таку діяльність коливається від 50 центів до 1 долара за 1000 дзвінків [65].

## 2.2 Типи даних, що використовуються для виявлення шахрайства

### 2.2.1 TAP3 та NRTRDE

Для того, щоб абонент мав можливість зробити дзвінок до іншої країни або користуватися послугами за межами своєї країни, оператори інфокомунікацій обговорюють та складають між собою домовленості [8]. Для обміну платіжної інформації між оператори зв’язку були розроблені стандарти TAP3 та NRTRDE.

**TAP3 (Transferred Account Procedures) –** це формат обміну CDR між мобільними операторами, що використовується у сценаріях роумінгу. Цей формат був розроблений GSMA у 1991 році та ув’являє собою файл з жосткою специфікацією та форматування, що використовує ASN1 для кодування даних. Існує два типи записів TAP3 - Notification Record and transferBatch.

Notification Record використовуються мережею оператора Б, щоб проінформувати оператора А, що вона доступна, але роумінг абоненти не використовують сервіси [35]. Цей тип записів надає інформацію про час створення файлу та час коли файл став доступним для відправки (file available/creation time), порядковий номер файлу (file sequence number – це

монотоно зростаюча цифра, що дозволяє отримувачу дізнатися, якщо якісь файли були загублені між файлом, що обробляється у теперешній час, та минулим файлом), час у який була вибрана певна кількість CDR на передачу (Transfer Cut Off Timestamp) та TAGID код отримувача та відправника (рис. 2.7).


```
{
  "notification": {
    "fileAvailableTimeStamp": {
      "localTimeStamp": "2023/03/21  15:44:18",
      "utcTimeOffset": "-0900"
    },
    "fileCreationTimeStamp": {
      "localTimeStamp": "2023/03/21  15:44:18",
      "utcTimeOffset": "-0900"
    },
    "fileSequenceNumber": "00003",
    "recipient": "RECV",
    "releaseVersionNumber": 12,
    "sender": "SEND",
    "specificationVersionNumber": 3,
    "transferCutOffTimeStamp": {
      "localTimeStamp": "2023/03/21  15:44:18",
      "utcTimeOffset": "-0900"
    }
  }
}
```


Рисунок 2.7 – Склад Notification Record у форматі JSON

transferBatch складається з файлового рівня та рівня CDR [36]. Рівень файлу складається з наступних частин:

- Batch Control Information, який містить TADIG код отримувача та відправника, порядковий номер файлу, Transfer Cut Off Timestamp, File

Available Timestamp, поля ідентифікації TAP3 (TAP3 identification fields) – номер версії специфікації (Specification Version Number) та номер версії випуску (Release Version Number);

- Accounting Information надає інформацію про податки та локальну валюту у CDR;
- Network Information містить інформацію стосовно мережевих елементів, що використовувалися під час надання послуг;
- Audit Control Information – надає повну інформацію про використаний сервіс та кошти користувачів у роумінгу на мережі Б з часами першої та останньої сесії.

Таблиця 2.2 – складові Batch Control Information

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Sender (DSP) | TADIG код відправника | |
| Recipient (ARP) | TADIG код отримувача | |
| File Sequence Number | Порядковий номер файлу | |
| Transfer Cut Off Timestamp (в тому числі UTC Time Offset) | Час у який була вибрана певна кількість CDR на передачу | |
| File Available Timestamp (в тому числі UTC Time Offset) | Час, коли файл був відправлений | |
| Specification Version Number | Значення 3 | |
| Release Version Number | Значення 12 | |

Таблиця 2.3 – складові Accounting Information

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Tax Rate Code | Повторювати з унікальним кодом до | Поле необхідне у випадку наявності податків |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | податкової ставки у записах дзвінків | |
| Tax Type | Значення 01 | Припустимо, що застосовується національний податок (якщо застосовується) |
| Tax Rate | Набуває значення використаної податкової ставки | |
| Tax Indicator | Значення 1 | Припустимо, що застосовується Value Added Tax. Якщо інший тип податку, поле не заповнюється |
| Local Currency | ISO код валюти, що використовується у розрахунку | Наприклад, EUR для євро |
| TAP Currency | У звичайному випадку, встановлюється Local Currency як частина домовленності між DSP/ARP | У звичайному випадку, конвертація валют не проводиться, але залежить від домовленності між DSP/ARP |
| Exchange Rate Code | Встановлюється унікальне значення | |
| Number of Decimal Places | Значення 6 або 0 | 6 – якщо проводиться конвертація валюти, у інших випадках 0 |
| Exchange Rate | Заповнюється відповідним значенням обміну валют | 1, якщо конвертація валюти не потрібна |

| Назва поля | Як заповнюється | Коментарі |
|---|---|---|
| TAP Decimal Places | Встановлюється унікальне значення | |

Таблиця 2.4 – складові Network Information

| Назва поля | Як заповнюється | Коментарі |
|---|---|---|
| UTC Time Offset Code | Повторювати з унікальним кодом до кожної унікального UTC Time Offset у записах дзвінків | |
| UTC Time Offset | Повторювати з унікальним кодом до кожної унікального UTC Time Offset у записах дзвінків | |
| Recording Entity Code | Повторювати з унікальним кодом до кожної унікального Recording Entity у записах дзвінків | |
| Recording Entity Type | Встановлюється значення в залежності від типу Recording Entity | |
| Recording Entity Identification | Повторювати з унікальним кодом до кожної унікального Recording | Заповнювати з справжнім Recording Entity Identification, якщо є у наявності. Якщо нема в наявності, то заповнюється як |

| | | |
|---|---|---|
| | Entity у записах дзвінків | “UNKNOWN” або VPMN TADIG код, якщо необхідна IP адреса, то заповнюється як “1.1.1.1” |

Таблиця 2.5 – складові Audit Control Information

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Total Charge | Підраховується з індивідуального запису дзвінків | |
| Total Tax Value | Підраховується з індивідуального запису дзвінків | |
| Total Discount Value | Значення 0 | |
| Call Event Details Count | Підраховується з індивідуального запису дзвінків | |

TAP3 рівень CDR може містити інформацію про вхідні та вихідні дзвінки, додаткові служби, вхідні та вихідні дзвінки VoLTE, GPRS та SMS події.

Таблиця 2.6 – складові вхідного дзвінку на рівні CDR

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| IMSI | IMSI | Якщо MSISDN не заповнене, то значення IMSI заповнюється повністю. Якщо MSISDN заповнене, то заповнення IMSI залишається на розсуд DSP (як мінімум MCC/MNC, що належать HPMN/DSP |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | | повинні бути заповнені) |
| MSISDN | MSISDN, якщо є в наявності | |
| Called number | Викликаємий номер, якщо значення в наявності | Для вхідних SMS, може приймати значення адреси SMSC |
| Dialed Digits | Набрані цифри, якщо є в наявності | |
| SMS Destination Number | Цільовий номер SMS, якщо є в наявності | |
| Call Event Start Timestamp (including UTC Time Offset) | Початок сесії дзвінка (в тому числі UTC Time Offset) | |
| Total Call Event Duration | Загальна тривалість сесії дзвінка | |
| Cause for Termination | Причина закінчення сесії | |
| Recording Entity Code | Числовий код | Заповнюється так само як Recording Entity Code у Network Information |
| Call Reference | Call Reference або Message Reference, якщо є в наявності | |
| Serving Network | TADIG код VPMN | Для антітромбування та зворотнього виклику, це можеть бути TADIG код HPMN. Це може бути використане як ознака, що сталося |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | | тромбування або зворотній виклик. |
| IMEI | IMEI, якщо є в наявності | |
| TeleService Code | Код TeleService, якщо є в наявності | Один з кодів TS чи BS повинні бути в наявності |
| Bearer Service Code | Код Bearer Service, якщо є в наявності | Один з кодів TS чи BS повинні бути в наявності |
| User Protocol Indicator | User Protocol Indicator, якщо є в наявності | Використовується для VT, але не є обов'язковим, так як BS37 може бути використаним. |
| Charged Item | Заповнюється з A, D, E та/або F в залежності під послуги за принципу нарахування | Якщо фіксовані компоненти були нараховані (наприклад, за нерегульовані послуги), тоді уся група Charge Information повинна бути повторена |
| Exchange Rate Code | Заповнюється відповідним кодом. | Заповнюється так само як Exchange Rate Code у Account Information |
| Call Type Level 1 | Заповнюється в залежності від типу нарахування. 1 - національні дзвінки, 2 - міжнародні дзвінки | Примітка: Call Type Level не копіюються з вхідного файлу TAP. Вони повинні |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | | відповідати типу нарахування у DSP |
| Call Type Level 2 | Заповнюється в залежності від типу нарахування, наприклад 0 - невідомий | 0 – невідомий<br>1- мобільний<br>2- ТМЗК<br>3 – Без географії<br>4 - Преміум тип нарахування<br>5 – супутникове призначення |
| Call Type Level 3 | Необхідно заповнюється, щоб унікально ідентифікувати тип нарахування | |
| Charge Type | Заповнюється як 00 | Припустимо, що повна вартість необхідна. Якщо є необхідність розірвати роумінг та вартість за з'єднання окремо (наприклад для SMS), тоді 01 та 03 можуть використовуватися як додаток до 00 у повторенні групи Charge Detail |
| Charge | Відповідно до типу розрахунку | Виключно без податків |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Chargeable Units | Заповнюйте у відповідності з Charged Item. Для Charged Item=D, заповнюйте з тривалістю в секундах. | |
| Tax Rate Code | Заповнюйте з відповідним кодом | Заповнюється так само як Tax Rate Code у Account Information Присутнє, якщо є податки |
| Tax Value | Заповнюйте з відповідним кодом | Присутнє, якщо є податки |
| HSCSD Indicator | Заповнюється як 1, якщо послуга є HSCSD | Використовується рідко |
| Supplementary Service Code | Якщо Supplementary Service Code є в наявності | Індикатор переадресації виклику |
| Third Party Number | Якщо Third Party Number є в наявності | |
| CLIR Status Indicator | Якщо CLIR Status Indicator є в наявності | |
| CAMEL Service Key | Якщо CAMEL Service Key є в наявності | |
| CAMEL Destination Number | Якщо CAMEL Destination Number є в наявності | |
| Operator Specific Information | Заповнюється додатковою інформацією як було домовлено (або на розсуд відправника) | Вільне тескстове поле. Може використовуватия для визначення |

| Назва поля | Як заповнюється | Коментарі |
|---|---|---|
| | | зворотнього виклику за допомогою використання спеціальних слів: Callback AntiTromboning |

Таблиця 2.7 – складові вихідного дзвінку на рівні CDR

| Назва поля | Як заповнюється | Коментарі |
|---|---|---|
| IMSI | IMSI | Якщо MSISDN не заповнене, то значення IMSI заповнюється повністю. Якщо MSISDN заповнене, то заповнення IMSI залишається на розсуд DSP (як мінімум MCC/MNC, що належать HPMN/DSP повинні бути заповнені) |
| MSISDN | MSISDN, якщо є в наявності | |
| Calling number | Номер, що викликає, якщо значення в наявності | Примітка: Це взагалі не є дуже надійним для TAP. Не всі HPMN надають його для індивідуального розрахунку |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| CLIR Status Indicator | CLIR Status Indicator, якщо є в наявності | |
| SMS Originator | Номер, що відправив SMS, якщо є в наявності | Якщо є в наявності та якщо SMS-MT є у складі |
| Call Event Start Timestamp (including UTC Time Offset) | Початок сесії дзвінка (в тому числі UTC Time Offset) | |
| Total Call Event Duration | Загальна тривалість сесії дзвінка | |
| Cause for Termination | Причина закінчення сесії | |
| Recording Entity Code | Числовий код | Заповнюється так само як Recording Entity Code у Network Information |
| Call Reference | Call Reference або Message Reference, якщо є в наявності | |
| Serving Network | TADIG код VPMN | |
| IMEI | IMEI, якщо є в наявності | |
| TeleService Code | Код TeleService, якщо є в наявності | Один з кодів TS чи BS повинні бути в наявності |
| Bearer Service Code | Код Bearer Service, якщо є в наявності | Один з кодів TS чи BS повинні бути в наявності |
| User Protocol Indicator | User Protocol Indicator, якщо є в наявності | Використовується для VT, але не є обов’язковим, так як |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | | BS37 може бути використаним. |
| Charged Item | Заповнюється з D, E та/або F в залежності під послуги за принципу нарахування | Якщо фіксовані компоненти були нараховані (наприклад, за нерегульовані послуги), тоді уся група Charge Information повинна бути повторена |
| Exchange Rate Code | Заповнюється відповідним кодом. | Заповнюється так само як Exchange Rate Code у Account Information |
| Call Type Level 1 | Заповнюється в залежності від типу нарахування. 1 - національні дзвінки, 2 - міжнародні дзвінки, 0 – для невідомого | |
| Call Type Level 2 | Заповнюється в залежності від типу нарахування, наприклад 0 - невідомий | 0 – невідомий<br>6 – переадресований дзвінок<br>7 - не переадресований дзвінок |
| Call Type Level 3 | Необхідно заповнюється, щоб унікально ідентифікувати тип нарахування | |
| Charge Type | Заповнюється як 00 | Припустимо, що повна вартість необхідна. |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | | Якщо є необхідність розірвати роумінг та вартість за з'єднання окремо (наприклад для SMS), тоді 01 та 03 можуть використовуватися як додаток до 00 у повторенні групи Charge Detail |
| Charge | Відповідно до типу розрахунку | Виключно без податків |
| Chargeable Units | Заповнюйте у відповідності з Charged Item. Для Charged Item=D, заповнюйте з тривалістю в секундах. | |
| Tax Rate Code | Заповнюйте з відповідним кодом | Заповнюється так само як Tax Rate Code у Account Information Присутнє, якщо є податки |
| Tax Value | Заповнюйте з відповідним кодом | Присутнє, якщо є податки |
| HSCSD Indicator | Заповнюється як 1, якщо послуга є HSCSD | Використовується рідко |
| CAMEL Service Key | Якщо CAMEL Service Key є в наявності | |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Operator Specific Information | Заповнюється додатковою інформацією як було домовлено (або на розсуд відправника) | Вільне тескстове поле. Може використовуватия для визначення антітромбування та зворотнього виклику за допомогою використання спеціальних слів: Callback AntiTromboning |

Таблиця 2.8 – складові додаткових служб (Supplementary Service Event) на рівні CDR

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| IMSI | IMSI | Якщо MSISDN не заповнене, то значення IMSI заповнюється повністю. Якщо MSISDN заповнене, то заповнення IMSI залишається на розсуд DSP (як мінімум MCC/MNC, що належать HPMN/DSP повинні бути заповнені) |
| MSISDN | MSISDN, якщо є в наявності | |
| Recording Entity Code | Числовий код | Заповнюється так само як Recording Entity Code у |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | | Network Information |
| Call Reference | Call Reference або Message Reference, якщо є в наявності | |
| Serving Network | TADIG код VPMN | Для антітромбування та зворотнього виклику, це можеть бути TADIG код HPMN. Це може бути використане як ознака, що сталося тромбування або зворотній виклик. |
| IMEI | IMEI, якщо є в наявності | |
| Supplementary Service Code | Якщо Supplementary Service Code є в наявності | Індикатор переадресації виклику |
| Action Code | Action Code | |
| Supplementary Service Parameters | Парметри додаткових служб, якщо є в наявності | |
| Charging Timestamp (including UTC Time Offset) | Час нарахування (містить у собі UTC Time Offset) | |
| Charged Item | Заповнюєься як E | |
| Exchange Rate Code | Заповнюється з відповідним кодом. | Заповнюється так само як Exchange Rate Code у Account Information |
| Charge Type | Заповнюється як 00 | |
| Charge | Відповідно до типу розрахунку | Виключно без податків |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Tax Rate Code | Заповнюйте з відповідним кодом | Заповнюється так само як Tax Rate Code у Account Information Присутнє, якщо є податки |
| Tax Value | Заповнюйте з відповідним кодом | Присутнє, якщо є податки |
| TeleService Code | Код TeleService, якщо є в наявності | |
| Bearer Service Code | Код Bearer Service, якщо є в наявності | |
| Operator Specific Information | Заповнюється додатковою інформацією як було домовлено (або на розсуд відправника) | Вільне тескстове поле. Може використовуватия для визначення антітромбування та зворотнього виклику за допомогою використання спеціальних слів: Callback AntiTromboning |

Таблиця 2.9 – складові дзвінка GPRS на рівні CDR

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| IMSI | IMSI | Якщо MSISDN не заповнене, то значення IMSI заповнюється повністю. Якщо MSISDN заповнене, то заповнення IMSI залишається на розсуд DSP (як мінімум |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | | HPMN/DSP повинні бути заповнені) |
| MSISDN | MSISDN, якщо є в наявності | |
| PDP Address | PDP адреса, якщо є в наявності | |
| Access Point Name NI | Access Point Name NI | |
| Access Point Name OI | Access Point Name OI, якщо є в наявності | |
| Call Event Start Timestamp (including UTC Time Offset) | Початок сесії дзвінка (в тому числі UTC Time Offset) | |
| Total Call Event Duration | Загальна тривалість сесії дзвінка | |
| Cause for Termination | Причина закінчення сесії | |
| Partial Type Indicator | Partial Type Indicator, якщо є в наявності | |
| PDP Context Start Timestamp (including UTC Time Offset) | PDP Context Start Timestamp (including UTC Time Offset), якщо є в наявності | Тільки для часткових записів |
| Network Init. PDP Context | Network Init. PDP Context if available, якщо є в наявності | |
| Charging Id | Ідентифікатор нарахування | |
| Recording Entity Code | Числовий код | Заповнюється так само як Recording Entity Code у Network Information |
| Serving Network | TADIG код VPMN | |
| IMEI | IMEI, якщо є в наявності | |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| IMS Signalling Context | Контекст IMS сигналізації, якщо є в наявності | |
| Data Volume Incoming | Обсяг вхідних даних | |
| Data Volume Outgoing | Обсяг вихідних даних | |
| Charged Item | Заповнюється за принципом нарахування | |
| Exchange Rate Code | Заповнюється відповідним кодом. | Заповнюється так само як Exchange Rate Code у Account Information |
| Call Type Level 1 | Заповнюється в залежності від маршрутизації | 10 HGGSN/HP-GW<br>11 VGGSN/VP-GW<br>12 Other GGSN/P-GW |
| Call Type Level 2 | Заповнюється в залежності від типу нарахування, наприклад 0 – невідомий або 10-15 для нарахування QoS (QoS charging) | 0 – невідомий<br>10- Широкосмуговий<br>11- Вузькосмуговий<br>12 – для розмов<br>13 - стрімінг<br>14 – Інтерактивний<br>15 – Background |
| Call Type Level 3 | Необхідно заповнюється, щоб унікально ідентифікувати тип нарахування | |
| Charge Type | Заповнюється як 00 | |
| Charge | Відповідно до типу розрахунку | Виключно без податків |
| Chargeable Units | Заповнюйте у відповідності з Charged Item. Для Charged | |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | Item=X, заповнюйте загальний обсяг у байтах. | |
| Tax Rate Code | Заповнюйте з відповідним кодом | Заповнюється так само як Tax Rate Code у Account Information Присутнє, якщо є податки |
| Tax Value | Заповнюйте з відповідним кодом | Присутнє, якщо є податки |
| CAMEL Service Key | Якщо CAMEL Service Key є в наявності | |
| Access Point Name NI | Якщо Access Point Name NI є в наявності | |
| Access Point Name OI | Якщо Access Point Name OI є в наявності | |
| Operator Specific Information | Заповнюється додатковою інформацією як було домовлено | Вільне текстове поле. |

Таблиця 2.10 – складові мобільної сесії на рівні CDR

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Mobile Session Service | Послуга мобільної сесії | 1 MO Voice over LTE<br>2 MT Voice over LTE<br>3 Emergency call over LTE |
| IMSI | IMSI | Якщо MSISDN не заповнене, то значення IMSI заповнюється повністю. Якщо MSISDN заповнене, то заповнення IMSI |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | | залишається на розсуд DSP (як мінімум MCC/MNC, що належать HPMN/DSP повинні бути заповнені) |
| MSISDN | MSISDN, якщо є в наявності | |
| Public User Id | Public User Id, якщо є в наявності | |
| IMEI | IMEI, якщо є в наявності | |
| Serving Network | TADIG код VPMN | Для антітромбування та зворотнього виклику, це можеть бути TADIG код HPMN. Це може бути використане як ознака, що сталося тромбування або зворотній виклик. |
| Event Reference | Event Reference | |
| Recording Entity Code | Числовий код | Заповнюється так само як Recording Entity Code у Network Information |
| Service Start Timestamp (including UTC Time Offset) | Початок сесії (в тому числі UTC Time Offset) | |
| Cause for Termination | Причина закінчення сесії | |
| Total Call Event Duration | Загальна тривалість сесії дзвінка | |
| Non-Charged Party Number | Non-Charged Party Number, якщо є в наявності | |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Non-Charged Public User Id | Non-Charged Public User Id, якщо є в наявності | |
| Requested Number | Requested Number, якщо є в наявності | |
| Requested Public User Id | Requested Public User Id, якщо є в наявності | |
| Charged Item | Заповнюється з A, D та/або F в залежності під послуги за принципу нарахування | Якщо фіксовані компоненти були нараховані (наприклад, за нерегульовані послуги), тоді уся група Charge Information повинна бути повторена |
| Exchange Rate Code | Заповнюється відповідним кодом. | Заповнюється так само як Exchange Rate Code у Account Information |
| Call Type Level 1 | Заповнюється в залежності від типу нарахування. 1 - національні дзвінки, 2 - міжнародні дзвінки | Примітка: Call Type Level не копіюються з вхідного файлу TAP. Вони повинні відповідати типу нарахування у DSP |
| Call Type Level 2 | Заповнюється в залежності від типу нарахування, наприклад 0 - невідомий | 0 – невідомий<br>1- мобільний<br>2- ТМЗК<br>3 – Без географії<br>4 - Преміум тип нарахування<br>5 – супутникове призначення |
| Call Type Level 3 | Необхідно заповнюється, щоб унікально ідентифікувати тип нарахування | |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Charge Type | Заповнюється як 00 | Припустимо, що повна вартість необхідна. |
| Charge | Відповідно до типу розрахунку | Виключно без податків |
| Chargeable Units | Заповнюйте у відповідності з Charged Item. Для Charged Item=D, заповнюйте з тривалістю в секундах. | |
| Tax Rate Code | Заповнюйте з відповідним кодом | Заповнюється так само як Tax Rate Code у Account Information Присутнє, якщо є податки |
| Tax Value | Заповнюйте з відповідним кодом | Присутнє, якщо є податки |
| Operator Specific Information | Заповнюється додатковою інформацією як було домовлено (або на розсуд відправника) | Вільне тескстове поле. Може використовуватия для визначення антітромбування та зворотнього виклику за допомогою використання спеціальних слів: Callback AntiTromboning |

Таблиця 2.11 – складові сервісу повідомлень на рівні CDR

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Messaging Event Service | Послуга мобільної сесії | 1 MO SMS over IP<br>2 MT SMS over IP<br>3 Emergency call over LTE |
| IMSI | IMSI | Якщо MSISDN не заповнене, то значення IMSI |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | | MSISDN заповнене, то заповнення IMSI залишається на розсуд DSP (як мінімум MCC/MNC, що належать HPMN/DSP повинні бути заповнені) |
| MSISDN | MSISDN, якщо є в наявності | |
| Public User Id | Public User Id, якщо є в наявності | |
| IMEI | IMEI, якщо є в наявності | |
| Serving Network | TADIG код VPMN | Для антітромбування та зворотнього виклику, це можеть бути TADIG код HPMN. Це може бути використане як ознака, що сталося тромбування або зворотній виклик. |
| Event Reference | Event Reference | |
| Element Type | Тип елементу, якщо є в наявності | Наприклад SMSC |
| Element ID | ІД елементу, якщо є в наявності | Наприклад адреса SMSC |
| Recording Entity Code | Числовий код | Заповнюється так само як Recording Entity Code у Network Information |
| Service Start Timestamp (including UTC | Початок сесії (в тому числі UTC Time Offset) | |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Time Offset) | | |
| Non-Charged Party Number | Non-Charged Party Number, якщо є в наявності | |
| Non-Charged Public User Id | Non-Charged Public User Id, якщо є в наявності | |
| Exchange Rate Code | Заповнюється відповідним кодом. | Заповнюється так само як Exchange Rate Code у Account Information |
| Call Type Level 1 | Заповнюється в залежності від типу нарахування. 1 - національні дзвінки, 2 - міжнародні дзвінки | Примітка: Call Type Level не копіюються з вхідного файлу TAP. Вони повинні відповідати типу нарахування у DSP |
| Call Type Level 2 | Заповнюється в залежності від типу нарахування, наприклад 0 - невідомий | 0 – невідомий<br>1- мобільний<br>2- ТМЗК<br>3 – Без географії<br>4 - Преміум тип нарахування<br>5 – супутникове призначення |
| Call Type Level 3 | Необхідно заповнюється, щоб унікально ідентифікувати тип нарахування | |
| Charge | Відповідно до типу розрахунку | Виключно без податків |
| Tax Rate Code | Заповнюйте з відповідним кодом | Заповнюється так само як Tax Rate Code у Account Information Присутнє, якщо є |

| Назва поля | Як заповнюється | Коментарі |
|---|---|---|
| | | податки |
| Tax Value | Заповнюйте з відповідним кодом | Присутнє, якщо є податки |
| Operator Specific Information | Заповнюється додатковою інформацією як було домовлено (або на розсуд відправника) | Вільне текстове поле. |

За умовами документа стандарту TD.104, записи TAP3 надаються до мережі А протягом 30 днів після закінчення дзвінка, інакше з мобільного оператора А не буде стягнена плата за використання послугами абонента на мережі Б. Якщо на мережі Б є проблеми у наданні записів TAP3 (операційні проблеми, перевірки, оновлення тарифікації, передача за допомогою MVNO, та т.п.), то час доставки збільшується до 40 днів. Для передачі TAP3 файлів, оператор А та Б можуть використовувати спільний FTP сервер або інший попередньо узгоджений варіант передачі, як наприклад через DCH.

**NRTRDE (Near Real Time Roaming Data Exchange) -** це формат обміну CDR між мобільними операторами, що використовується для захисту від шахрайства та перевірки доходів. Цей формат також використовує ASN1 для кодування інформації [37,38]. Складається NRTRDE файл з наступних частин:

- Рівень файлу (File Level), який містить TADIG код отримувача та відправника, порядковий номер файлу, кількість подій викликів (Call Events Count), File Available Timestamp, поля ідентифікації NRTRDE (NRTRDE identification fields) – номер версії специфікації (Specification Version Number) та номер версії випуску (Release Version Number);
- Рівень деталізації дзвінка (Call Detail Level), що може містити інформацію про вхідні та вихідні дзвінки, GPRS

Таблиця 2.12 – складові файлового рівню

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Specification Version Number | Значення 2 | |
| Release Version Number | Значення 1 | |
| Sender | TADIG код відправника (DSP) | |
| Recipient | TADIG код отримувача (ARP) | |
| Sequence Number | Порядковий номер файлу | |
| File Available Timestamp (including UTC Time Offset) | Час, коли файл був відправлений | |
| Call Events Count | Підраховується з індивідуальних записів дзвінків | |

Таблиця 2.13 – складові вихідного дзвінка у рівні деталізації дзвінка

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| IMSI | IMSI | Якщо MSISDN не заповнене, то значення IMSI заповнюється повністю. Якщо MSISDN заповнене, то заповнення IMSI залишається на розсуд DSP (як мінімум MCC/MNC, що належать HPMN/DSP повинні бути заповнені) |
| IMEI | IMEI, якщо є в наявності | |
| Call Event Start Timestamp (including UTC Time Offset) | Початок сесії дзвінка (в тому числі UTC Time Offset) | |
| Total Call Event | Загальна тривалість сесії | |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Duration | дзвінка | |
| Cause for Termination | Причина закінчення сесії | |
| TeleService Code | Код TeleService, якщо є в наявності | Один з кодів TS чи BS повинні бути в наявності |
| Bearer Service Code | Код Bearer Service, якщо є в наявності | Один з кодів TS чи BS повинні бути в наявності |
| Supplementary Service Code | Якщо Supplementary Service Code є в наявності | Індикатор переадресації виклику |
| Dialed Digits | Набрані цифри, якщо є в наявності | |
| Connected Number | З'єданий номер, якщо є в наявності | Для SMS-MO, це буде адреса SMSC |
| Third Party Number | Якщо Third Party Number є в наявності | |
| Recording Entity Identification | Recording Entity Identification | Заповнюється справжнім Recording Entity Identification при нявності. Якщо відсутній, то заповнюється як UNKNOWN або TADIG код, що належить VPMN |
| Call Reference | Call Reference або Message Reference, якщо є в наявності | |
| Serving Network | TADIG код VPMN | Для антітромбування та зворотнього виклику, це можеть бути TADIG код HPMN. Це може |

| Назва поля | Як заповнюється | Коментарі |
|---|---|---|
| | | бути використане як ознака, що сталося тромбування або зворотній виклик. |
| MSISDN | MSISDN | Заповнюється, якщо є в наявності |

Таблиця 2.14 – складові вхідного дзвінка у рівні деталізації дзвінка

| Назва поля | Як заповнюється | Коментарі |
|---|---|---|
| IMSI | IMSI | Якщо MSISDN не заповнене, то значення IMSI заповнюється повністю. Якщо MSISDN заповнене, то заповнення IMSI залишається на розсуд DSP (як мінімум MCC/MNC, що належать HPMN/DSP повинні бути заповнені) |
| IMEI | IMEI, якщо є в наявності | |
| Call Event Start Timestamp (including UTC Time Offset) | Початок сесії дзвінка (в тому числі UTC Time Offset) | |
| Total Call Event Duration | Загальна тривалість сесії дзвінка | |
| Cause for Termination | Причина закінчення сесії | |
| TeleService Code | Код TeleService, якщо є в наявності | Один з кодів TS чи BS повинні бути в наявності |
| Bearer Service | Код Bearer Service, якщо є в | Один з кодів TS чи BS |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Code | наявності | повинні бути в наявності |
| Calling Number | Номер абоненту, що викликає, якщо є в наявності | Для SMS-MT, це буде адреса SMSC |
| Recording Entity Identification | Recording Entity Identification | Заповнюється справжнім Recording Entity Identification при нявності. Якщо відсутній, то заповнюється як UNKNOWN або TADIG код, що належить VPMN |
| Call Reference | Call Reference або Message Reference, якщо є в наявності | |
| Serving Network | Якщо, значення Serving Network присутне у вхідній записі, тому заповнюйте цим значенням.<br>Інакше, заповнюйте значенням TADIG коду VPMN | |
| MSISDN | MSISDN | Заповнюється, якщо є в наявності |

Таблиця 2.15 – складові GPRS у рівні деталізації дзвінка

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| IMSI | IMSI | Якщо MSISDN не заповнене, то значення IMSI заповнюється повністю. Якщо MSISDN заповнене, то заповнення |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| | | розсуд DSP (як мінімум MCC/MNC, що належать HPMN/DSP повинні бути заповнені) |
| IMEI | IMEI, якщо є в наявності | |
| Call Event Start Timestamp (including UTC Time Offset) | Початок сесії дзвінка (в тому числі UTC Time Offset) | |
| Total Call Event Duration | Загальна тривалість сесії дзвінка | |
| Cause for Termination | Причина закінчення сесії | |
| Access Point Name NI | Access Point Name NI | |
| Access Point Name OI | Access Point Name OI, якщо є в наявності | |
| Data Volume Incoming | Обсяг вхідних даних | |
| Data Volume Outgoing | Обсяг вихідних даних | |
| SGSN Address | Адреса SGSN | Заповнювати з справжньою адресою SGSN, якщо є у наявності. Якщо нема в наявності, то заповнюється як “1.1.1.1” |
| GGSN Address | Адреса GGSN | Заповнювати з справжньою адресою GGSN, якщо є у наявності. Якщо нема в наявності, то заповнюється як “1.1.1.1” |

| **Назва поля** | **Як заповнюється** | **Коментарі** |
|---|---|---|
| Charging Id | Ідентифікатор нарахування | |
| Serving Network | Якщо, значення Serving Network присутне у вхідній записі, тому заповнюйте цим значенням.<br>Інакше, заповнюйте значенням<br>TADIG коду VPMN | |
| MSISDN | MSISDN, якщо є в наявності | |

За умовами документа стандарту TD.106, записи NRTRDE надаються до мережі А протягом 4 годин після закінчення дзвінка. Якщо на мережі Б є проблеми у наданні записів NRTRDE (операційні проблеми, перевірки, оновлення тарифікації, передача за допомогою MVNO, та т.п.), то час доставки збільшується до 8 годин. Для передачі NRTRDE файлів, оператор А та Б можуть використовувати спільний FTP сервер або інший попередньо узгоджений варіант передачі, як наприклад через DCH [8].

### 2.2.2 Збір даних безпосередньо з мережі за допомогою мережевого зонду

Збір даних безпосередньо з мережі є найбільш привабливим варіантом, так як цей метод дозволяє отримати кожне повідомлення сигналізації в межах своей мережі, визначити якість наданих послуг та додатки, що використовують абоненти мережі [5].

Звичайно, що даний спосіб має свої недоліки та труднощі у реалізації. На прикладі Gigamon fabric solution та Huawei NetProbe розглянемо особливості реалізації та підготовки інформації з мережі для наступної обробки у аналітичних платформах Big data.

Як було наведено в минулому розділі, **Gigamon fabric solution** є частиною The Gigamon and Argyle Data Joint Solution (рис 2.8).

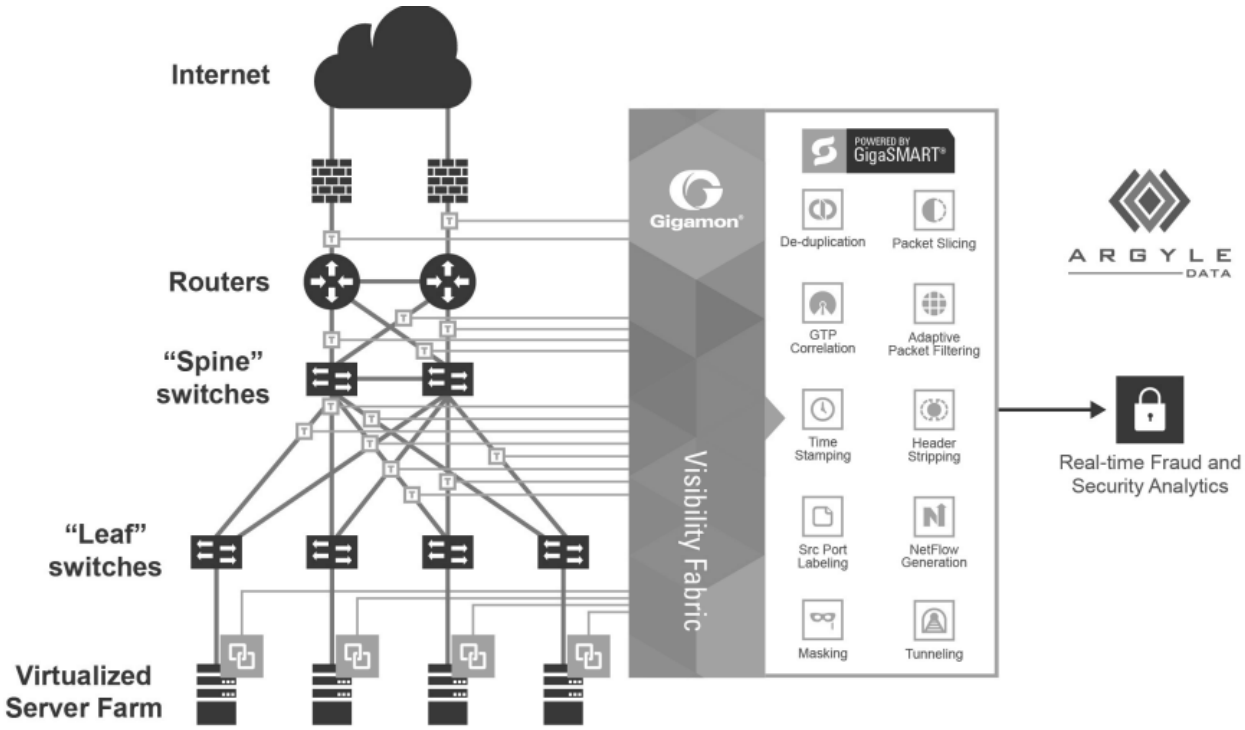



Рисунок 2.8 – Складові Gigamon fabric solution

Gigamon fabric solution оброблює трафік у наступній послідовності:

Спершу, кореляція між абонента та його трафіком досягається шляхом розгортання програми кореляції GTP GigaSMART від Gigamon. Протокол GPRS Tunneling Protocol (GTP) зазвичай використовується для передачі мобільних даних через мережі та включає трафік рівня сигналізації (control plane) та рівня даних користувача (user data plane). Відображення активності абонента вимагає здатність розуміння природу та станів трафіку GTP для співвідношення сесій з конкретним користувачем мережі, щоб отримати точне уявлення про його дії [21].

По-друге, запатентована Gigamon технологія Flow Mapping допомагає зменшити трафік для обробки, щоб підвищити продуктивність аналізатора. Flow Mapping — це технологія, яка міститься у вузлах GigaVUE Visibility Fabric від Gigamon, яка приймає трафік пропускної спроможності 1 ГБ, 10 ГБ, 40 ГБ або 100 ГБ від мережевого спліттера або порту SPAN/відзеркалення (фізичного чи віртуального), а потім оптимізує потоки на основі індивідуальних профілів трафіку за допомогою інструментів і програм, які забезпечують захист, моніторинг і аналіз інфраструктури CSP.

По третє, додаток Gigamon FlowVUE пропонує рішення группування трафіку абонента на основі IP, що дозволяє операторам перетворювати Big data на інформацію, якою можна керувати та застосовувати. Додаток дозволяє існуючим інструментам підключатися до високошвидкісних потоків, забезпечуючи репрезентативне уявлення про трафік для діагностики та візуалізації.

Нарешті, GigaSMART забезпечує Adaptive Packet Filtering, яка забезпечує потужний механізм фільтрації, який ідентифікує вміст у будь-якій частині пакету, включаючи корисне навантаження (payload) пакету. Цей механізм дає змогу фільтрувати на основі конкретних параметрів протоколу інкапсуляції, включаючи ідентифікатор тунелю GTP (GTP tunnel ID), ідентифікатор VXLAN, VN-Tag src/dst vif та багато іншого, включаючи трафік SSL. Крім того, оператори також мають можливість дивитися за межі протоколів інкапсуляції та переглядати оригінальний інкапсульований пакет, щоб фільтрувати за допомогою IP джерела/одержувача або за номерами портів 4 рівня моделі TCP/IP.

Gigamon fabric solution підтримує SS7, IP, мережі 3G, LTE, що у свою чергу дозволяє обробку наступних сервісів [43]:

- Мобільні дзвінки—ISUP,TD.35, MAP, SMS, BSSAP, GTP, Diameter, xCDR та інше
- Дзвінки з фіксованої мережі—ISUP, Diameter, xCDR та інше
- VoIP/VoLTE дзвінки—SIP, H.323, Diameter, xCDR та інше

**Huawei NetProbe** є частиною Huawei SmartCare та GENEX Discovery. NetProbe використовується для декодування та створення детального запису за час сеансу зв’язку, зберігання необробленної сигналізації та виконання трасування у режимі реального часу. Мережевий зонд може роспізнати більше ніж 1300 різних видів протоколів та та додатків за допомогою Service Awareness Engine, особистою розробкою компанії Huawei. NetProbe підтримує збір трафіку з мережей NGN, GSM (пакетна та канальна комутація), UMTS (пакетна та канальна комутація), LTE, IMS. База DPI (Deep protocol inspection), за допомогою якої мережевий зонд розпізнає дані, постійно оновлюється та дозволяє додавати та налаштовувати свої протоколи [10].

Таблиця 2.16 – приклад послуг/протоколів, що підтримує NetProbe.

| **Назва сервісу** | **Приклад послуг/протоколів, що підтримує система** |
|---|---|
| Базові сервіси | WAP1.X/2.0, HTTP/ HTTPS, Facebook, Twitter, Radius, Gaming, Win_Update та інші |
| Email | SMTP (SSL), POP3 (SSL), IMAP4 (SSL), Webmail, MS_Exchange, LotusNotes, Blackberry та інші |
| P2P | eDonkey, Bittorrent, FlashGet, Thunder, HotLine, GNUTELLA, DirectConnect та інші |
| VoIP | Skype Out/In, SIP, Diameter, H323, MGCP, Net2Phone, GoogleTalk, Shutter, UUCall та інші |
| Streaming | RTP/RTSP, RealPlayer, MS_Media, Flash_Yahoo, PPLIve, YouTube, AOL_Video та інші |
| IM | MSN, GoogleTalk, YahooMsg, Skype IM, ICQ, Viber, Whatsapp та інші |

Для пакетної мережі мержевий зонд наступним чином (рис 2.9) - спочатку система потік даних за допомогою інформації у таблиці (Flow Table Match), потім ідентифікує протоколи 3-4 рівня моделі OSI (L3/L4 Protocol Identification) та парсить їх (L3/L4 Parse). Наступним кроком, система розглядає та ідентифікує 7й рівень протоколів (L7/L7 Protocol Identification), що пройшли ідентифікацію на 3-4 рівні минулого кроку (L3/L4 Protocol Identification), роспізнає додатки та робить парсинг(L7/L7+ Apps Parse). Після чого NetProbe відправляє результати парсингу протоколів 3-4 рівня (L3/L4 Parse) та парсингу та роспізнання додатків на 7-му рівні (L7/L7+ Apps Parse) до серверів медіації (MES) та розподілення обробки (DPS) систем GENEX Discovery або SEQ Analyst. Для мережі з канальною комутацією каналів, спочатку сигналізація декодується, а потом генерюється детальзований запис для кожної сесії та кожного дзвінка.

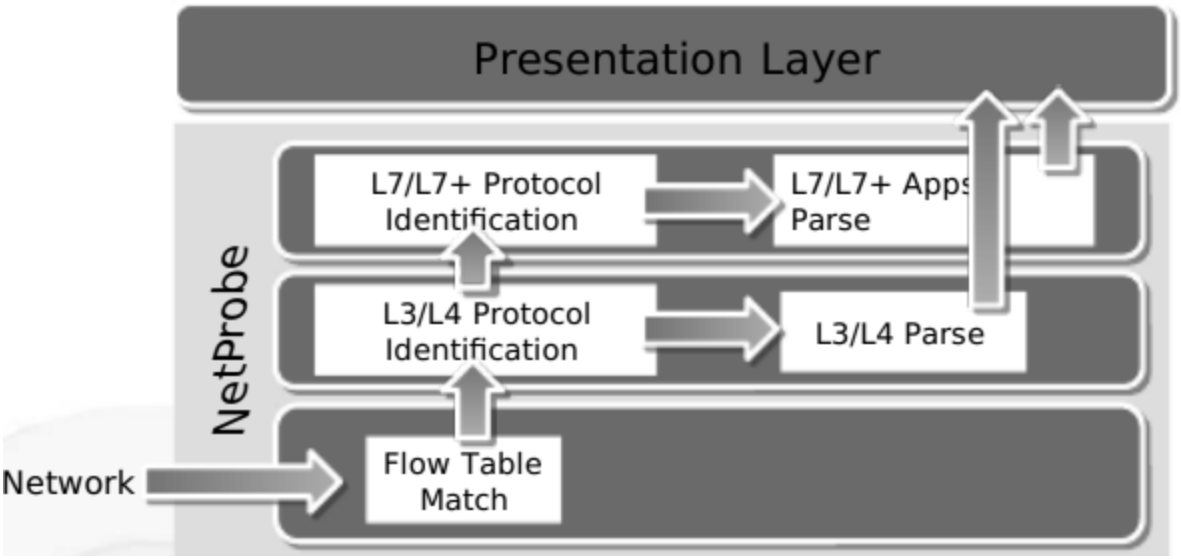



Рисунок 2.9 – Принцип роботи Huawei NetProbe

Як і Gigamon fabric solution, NetProbe використовує сплітери мережеві відгалужувачі для отримання інформації. Система підтримує потоки пропускної спроможності 1 ГБ, 10 ГБ, 40 ГБ або 100 ГБ (рис 2.10).

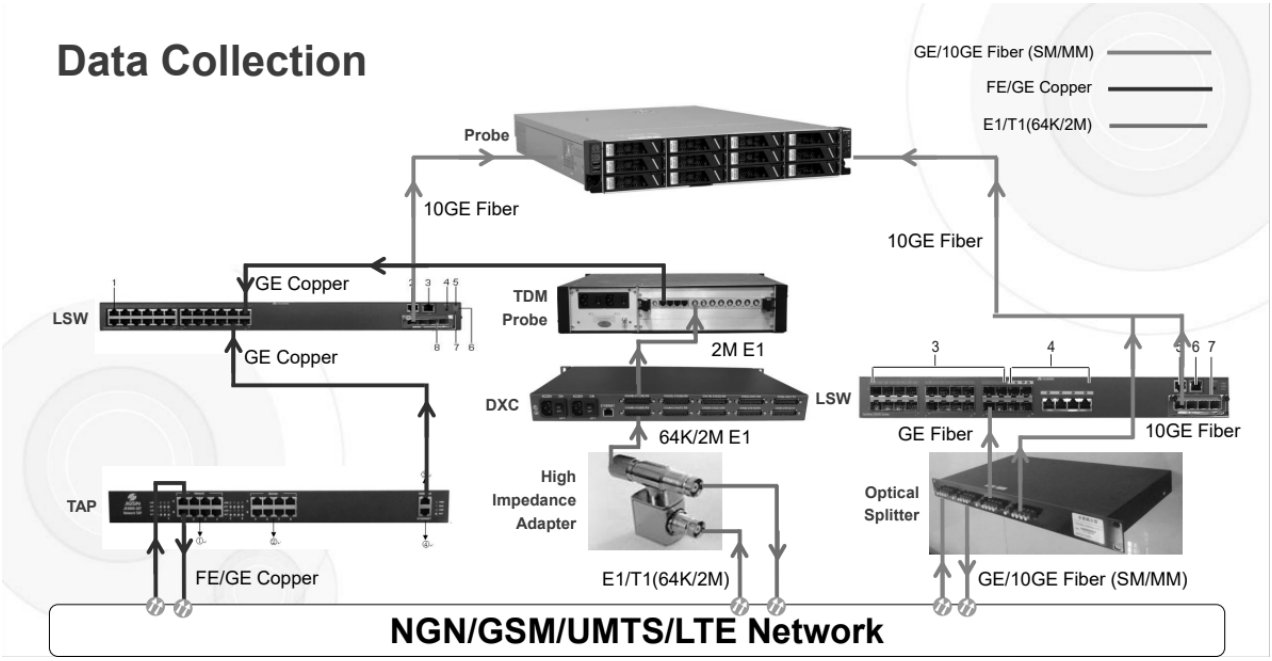



Рисунок 2.10 – Методи збору Huawei NetProbe

Сплітери та мережеві відгалужувачі теж встановлюються між елементами мережі оператора інфокомунікацій (рис. 2.11). Ці пристрої з’єднуються NeProbe за

допомогою мережеві комутатори (LSW) на прикладі наведеному на рисунку 2.8. Скоріш за все, ці комутатори також виконують функції фільтрування дублювання пакетів та трафіку мережі, бо NetProbe не має такого функціоналу.

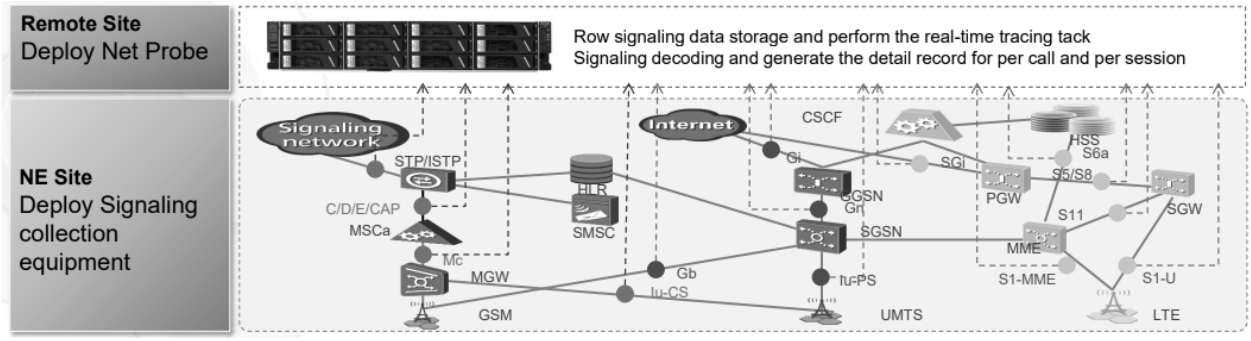



Рисунок 2.11 – Приклад встановлення обладнання на мобільної мережі

Обидви системи для збору трафіку використовуються зеркалювання трафіка з портів або мережевий відгалужувач (спліттер).

Мережевий відгалужувач (network TAP (Test Access Point)) – це фізичний пристрій, що під'єднується безпосередньо до кабельної інфраструктури, щоб розділити або скопіювати трафік для використання у аналітиці, безпеці або для загального керування мережею [22]. На рисунку 2.12 наведено принцип роботи.

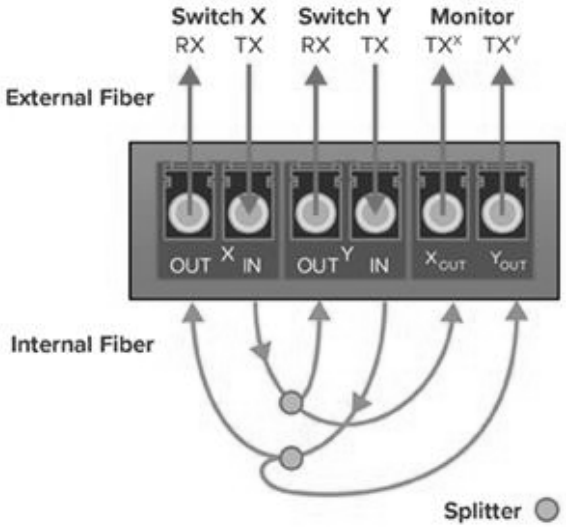



Рисунок 2.12 – Схема роботи пасивного відгалужувача

У першу пару портів входить оптичний кабель RX/TX (receive/transfer) від комутатора X (Switch X), а у другу пару портів оптичний кабель RX/TX від

комутатора Y (Switch Y), а третя пара портів це відгалужений TX від обох комутаторів. Таке відгалуження відбувається за рахунок зменшення рівня сигнала, тому при встановці цих пристроїв необхідно розрахувати ймовірні втрати по рівню сигнала.

Таблиця 2.17 – максимальні втрати сигналу в залежності від коефіцієнта розподілу

| Мультімодовий пасивний | | | |
|---|---|---|---|
| Коефіцієнт розподілу | 50/50 | 60/40 | 70/30 |
| Максимальні мережеві втрати | 3.9dB | 3.15dB | 2.2dB |
| Максимальні втрати на моніторинг | 3.9dB | 5.15dB | 6.2dB |
| Синглмодовий пасивний | | | |
| Коефіцієнт розподілу | 50/50 | 60/40 | 70/30 |
| Максимальні мережеві втрати | 3.7dB | 3.05dB | 2.0dB |
| Максимальні втрати на моніторинг | 3.7dB | 4.95dB | 6.1dB |

Існуть 2 типи таких пристроїв – активні та пасивні. Відмінність активного пристрою від пасивного полягає у тому, що він може регенувати рівень сигналу та потребує живлення для своєї роботи, що робить його менш надійним у порівнянні з пасивним відгалужувачем.

Зеркалювання порту – це функціонал, що дублює трафік з одного порту комутатора на інший порт (рис. 2.13). Зазвичай, такий функціонал використовується для визначення розташування несправності. При використані

даної функції, після виявлення несправності виробники обладнання дають рекомендацію вимкнути її. При довготривалому користуванні може виникнути падіння працездатності пристрою, що може вплинути на інші сервіси, що налаштовані та надає комутатор. Тому необхідно слідкувати чи є достатньо ресурсів для довготривалого використання зеркалювання.

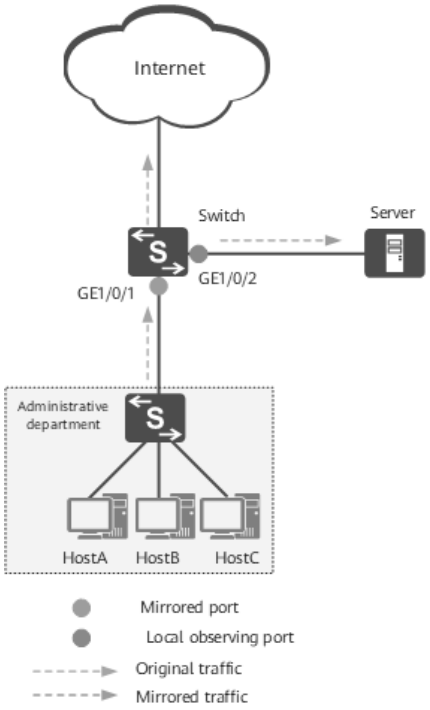



Рисунок 2.13 – Схема зеркалювання порту

Окрім наведених вище засад при використанні першої чи другої технології збору інформації мережного зонду, в обох наведених схемах представлені приклади мереж, основні елементи якої знаходяться в межах одного об’єкту. Зазвичай оператори інфокомунікацій розподіляють навантаження обробки наданих послуг та роблять резервування між декількома об’єктами за допомогою віртуалізації мережевих функцій (VNF – virtualized network functions) [5]. У випадку, якщо базова мережа LTE, наприклад, що знаходиться у місті А видає відмову у обслуговуванні завдяки по якийсь причині, то її навантаження повинна взяти базова мережа LTE, що знаходиться у місті Б (рис 2.14).

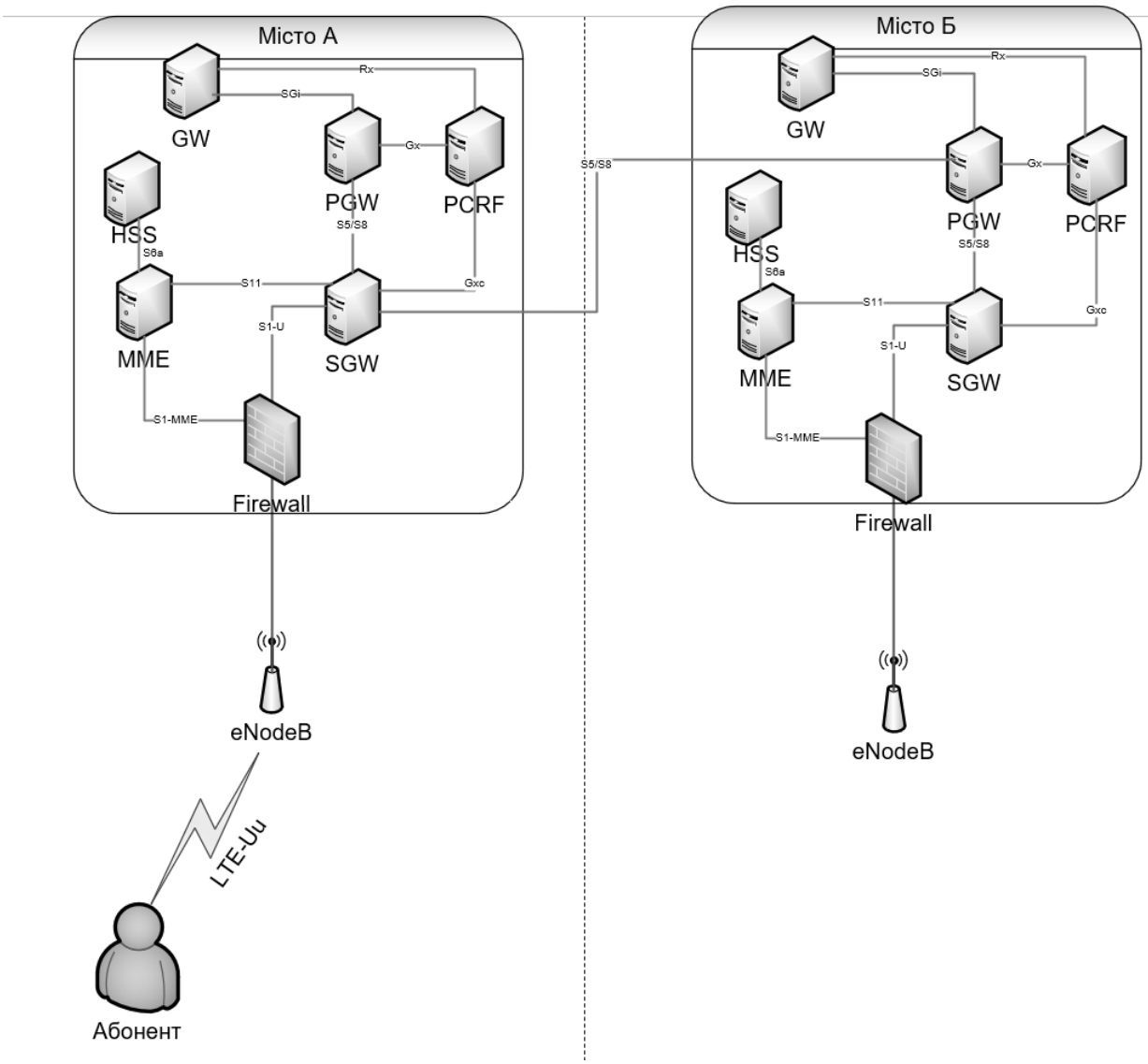



Рисунок 2.14 – Приклад мережі LTE на основі VNF

Під час резервування віртуалізації мережевих функцій було виявлено наступні складності:

- Додаткові витрати на обслуговування - додаткові мережові розгалужовачі повинні бути встановлені у місті Б, щоб була можливість встановити повний моніторинг даних замість частково. У випадку оновлення апаратної частини, встановлення нових елементів або навіть зміни одного виробника на іншого, повинні бути встановлені нові розгалужовачі або ж перез'єднані існуючі в залежності від можливостей та специфікацій нових елементів;

- Дублювання даних під час взаємодії з резервним середовищем: з’являється сценарії дублювання даних, коли абонента у місті А обслуговує SGW у цьому ж місті, доки обслуговуючий його PGW знаходиться у місті Б. У такому випадку система моніторингу отримує дублюючі деталізові записи, що може засягати 33% від усіх записів;
- Встановлення додаткового мережного зонду та оновлення ліцензії: якщо один мережевий зонд буде збирати інформацію з міста А та Б, то є ризик збільшення навантаження магістральної мережі у 2 рази, бо трафік базової мережі може коливатися від 1 до 100 ГБ/с і більше. Оновлення ліцензії теж потребується, бо кількість трафіку, що може обробити мережа теж зростає на 50%.

Витрати на обслуговування мережі розраховується за допомогою моделі підтримки її інфраструктури[].

$$Cost_{operation} = R\left(\mathrm{M}_{total} * \mathrm{S}_{avg} + \mathrm{IT}_{dep} + \sigma_l\right) \quad (2.1)$$

Дана модель витрат (формула 2.1) показує залежність кількості повних шаф з обладнанням R від кількості IT персоналу $\mathrm{M}_{total}$ та їх середньої заробітної плати на місяць $\mathrm{S}_{avg}$, депривації обладнання у шафі з часом $\mathrm{IT}_{dep}$, витратами на ліцензювання апаратної частини сервера та ОС з урахуванням встановлення патчів.

$$\mathrm{M}_{total} = \frac{1}{N_R} \quad (2.2)$$

Де $N_R$ кількість шаф на одного спеціаліста.

$$\mathrm{IT}_{dep} = \frac{P_{rprice}}{T_{rlife}} \quad (2.3)$$

$P_{rprice}$ ціна на придбання шафи з апаратною частиною, $T_{rlife}$ життєвий цикл шафи з обладнанням.

$$\sigma_l = \frac{P_{ltotal}}{R} \quad (2.4)$$

$P_{ltotal}$ загальна ціна на ліцензування.

З урахуванням виявлених складнощів, формула була удосконалена.

$$Cost_{operation} = \sum_{i=1}^{R} \left( \mathrm{M}_{total} * \mathrm{S}_{avg} + \mathrm{IT}_{depi} + \sigma_{li} + \sigma_{swli} + \mathrm{IT}_{spli} \right) \quad (2.5)$$

Операційні витрати враховують кількісті шаф базової мережі $R$, $\mathrm{M}_{total}$ загальна кількість персоналу з урахуванням підтримки зонду, $\sigma_{swli}$ витрати на ліцензування ПЗ з урахуванням кількості трафіку мережі, $\mathrm{IT}_{spli}$ витрати використання розгалужувачів.

Для підтримки проби, окрім IT спеціаліста потрібен спеціаліст, що має знання базової мережі, комутаторів, мережевих зондів. Тобто ще 3 спеціаліста.

$$\mathrm{M}_{total} = \frac{1}{N_R} + \frac{3}{N_{project}} \quad (2.6)$$

Де $N_{project}$ кількість систем на одного спеціаліста.

$$\sigma_{swl} = \frac{P_{swtotal}}{T_{monlife}} \quad (2.7)$$

$P_{swtotal}$ ціна на придбання ліцензії ПЗ проби,$T_{monlife}$життєвий цикл шафи з обладнанням базової мережі.

$$\mathrm{IT}_{spl} = \frac{N_{monp} * P_{sprice}}{N_{splp} * T_{monlife}} \quad (2.8)$$

$N_{monp}$ загальна кількість фізичних портів на моніторинг, $N_{splp}$ кількість портів на сплітері, $P_{sprice}$ ціна одного розгалужувача.

## 2.3 Висновки до розділу

У першій частині даного розділі були розглянуті вплив шахрайства на інфокомунікаційну мережу, причини його виникнення та види шахрайства. Розглядаються найбільш види шахрайств у витратах, такі як International Revenue Share Fraud (IRSF), Arbitrage, Interconnect Bypass, Domestic Premium Rate Service, Traffic Pumping та більш прості або менші види шахрайства. Розглядаючі опис та порівнюючи ці типи між собою, було помічено що при систематизації один тип шахрайства може входити в інший тип або ці два типи можуть об’єднуватися в третій. Це штохває до висновку, що одна й та сама шахрайська активність може

бути визначена різними правилами виявлення. Також, відмечено що шахраї, окрім використання технічних обмежень у реалізації або технічних завадах у безпеці, за допомогою соціальної інженерії можуть отримати доступ до мережі оператора інфокомунікацій. Дане дослідження більш спрямове на технічне виявлення подій шахрайства, тому людській фактор виходить за рамки та повинен досліджуватися як окреме явище.

У другій частині даного розділі були розглянуті основні типи даних, що використовуються у виявленні шахрайства, а саме NRTRDE, TAP3 та дані мережі, які збираються за допомогою мережевих зондів. Було наведено складові NRTRDE та TAP3, що кодуються за допомогою ASN1. Цей тип даних є стандартизованим та використовується для обміну інформацією стосовно послуг, які абонент використав у роумінгу. Розглянуті типи сервісів що можуть передані від однієї мережі до іншої за допомогою NRTRDE та TAP3. Дані файли можуть бути передані не більше ніж 4 години від кінця сесії (NRTRDE) та не більше ніж 30 днів від кінця сесії (TAP3).

Детально описано методику збору мережевих даних за допомогою мережевих зондів. Проаналізовані складові та технології, що саме використовуються, а саме відгалужувачі трафіку, сплітери та зеркалювання портів. При використанні відгалужувачів трафіку або сплітерів треба мати на увазі, що вони встановлюються між елементів базової мережі та можуть мати вплив на сервісі у випадку неправильних розрахунків рівня сигналу. У випадку зеркалювання потрібно слідкувати за працездатністю пристрою, бо даний функціонал може викликати перевантаження при тривалому використанні без урахування сервісів, що працюють через пристрій.

Отримав подальшого розвитку метод моніторингу віртуалізованого середовища з резервуванням, який на відміну від існуючих дозволив виявити дублікацію даних, встановлення додаткового мережного зонду під час розширення мережі для удосконалення моделі підтримки її інфраструктури.

# 3. РОЗРОБКА ІНТЕРФЕЙСУ НА ОСНОВІ КОМПЛЕКСНОГО ВИКОРИСТАННЯ CDR З РІЗНИХ ДЖЕРЕЛ

## 3.1 Розробка ключових показників ефективності системи

У систем, розглянутих у минулих розділах, є 3 основні етапи - завантаження/збір даних (data loading/collection), аналітика інформації (data analysis) та створення оповіщення/справи про шахрайство (alert/case creation) (рис. 3.1) [4]. Щоб визначити продуктивність кожного етапу будемо використовувати $T_{coll}$ для завантаження/збору даних, $T_{det}$ для аналітика інформації, $T_{cd}$ для створення оповіщення/справи про шахрайство. Рекомендую збирати статистику за один день, щоб перевірити виявлення у проміжки годин низького трафіку (нічний час), під час робочого дня (високе користування мережею) та після робочого дня (вечірній час). Наступний підхід може бути використаний для кожного джерела даних окремо або разом, щоб побачити наскільки підвищиться виявлення шахрайства.

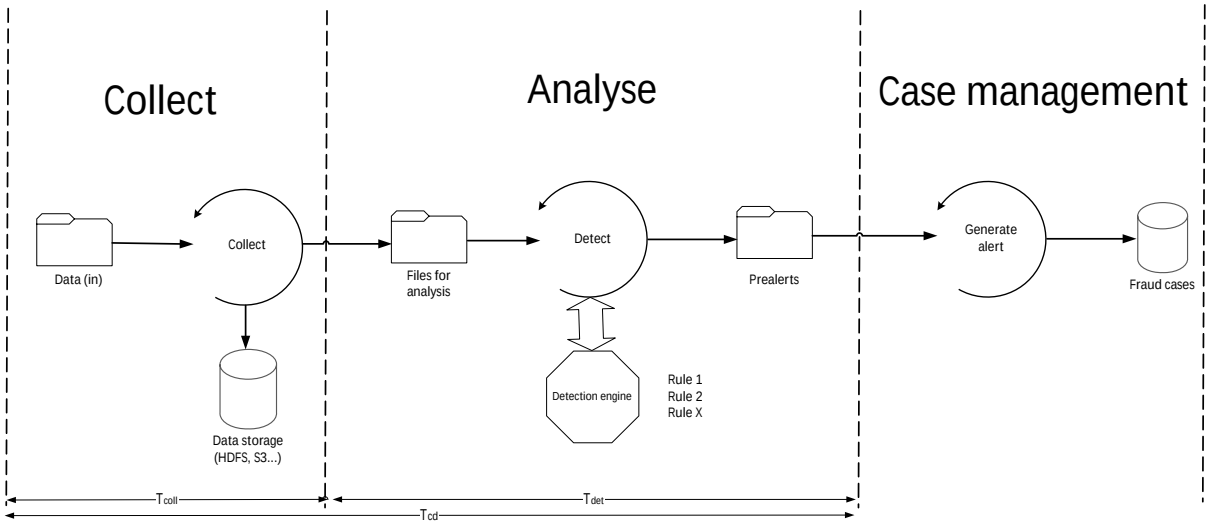



Рисунок 3.1 – Спільна схема систем виявлення шахрайства

$T_{coll}$ це різниця між часом кінця дзвінка або сесії ($T_{session\_end\_time}$) та часом, коли CDR був завантажений до сховища даних системи ($T_{insertion\_time}$).

$$T_{coll} = T_{session\_end\_time} - T_{insertion\_time} \quad (3.1)$$

$T_{det}$ це різниця між часом появи оповіщення про шахрайства для підозрілого суб'єкту ($T_{alert\_creation\_time}$) та часом коли CDR був завантажений до сховища для цього суб'єкту ($T_{suspect\_insertion\_time}$)

$$T_{det} = T_{alert\_creation\_time} - T_{suspect\_insertion\_time} \quad (3.2)$$

$T_{cd}$ це різниця між часом кінця дзвінка або сесії ($T_{session\_end\_time}$) та часом появи оповіщення про шахрайства для підозрілого суб'єкту ($T_{alert\_creation\_time}$).

$$T_{cd} = T_{alert\_creation\_time} - \ T_{session\_end\_time} \quad (3.3)$$

Для того щоб зробити масштабування та візуалізацію для часу необхідного для кожного етапу, можливо поділити отримані значення на інтервали по 5 хвилин для більш точної статистики. В залежності від результатів розрахунків, можливо використовувати інтервали по 10 хвилин або більше. Після розбиття розраховуємо середньозважене значення для кожного типу затримки.

Середньозважене значення $T_{coll}$

$$\frac{\sum_{i=1}^{n} T_{coll_i}\, a_i}{\sum_{i=1}^{n} a_i} \quad (3.4)$$

Середньозважене значення $T_{det}$

$$\frac{\sum_{i=1}^{n} T_{det_i}\, a_i}{\sum_{i=1}^{n} a_i} \quad (3.5)$$

Середньозважене значення $T_{cd}$

$$\frac{\sum_{i=1}^{n} T_{cd_i} a_i}{\sum_{i=1}^{n} a_i} \tag{3.6}$$

Де $a$ це кількість значень, що знаходяться у часовому інтервалі, а n це кількість тих самих інтервалів.

У випадку декількох джерел інформації для розрахунку використовуємо середнє значення на основі вагомого коефіцієнту. Для цього середнє значення для$T_{coll}$ , $T_{det}$, $T_{cd}$.

$$\tilde{T_{coll}} = \frac{\sum_{i=1}^{n} T_{coll_i}}{n} \tag{3.7}$$

$$\tilde{T_{det}} = \frac{\sum_{i=1}^{n} T_{det_i}}{n} \tag{3.8}$$

$$\tilde{T_{cd}} = \frac{\sum_{i=1}^{n} T_{cd_i}}{n} \tag{3.9}$$

Де n – загальна кількість оповіщень про шахрайство.

Наступним кроком рахуємо середнє значення затримки для усіх отриманих деталізованих записів кожного типу.

$$\tilde{T_{acoll}} = \frac{\sum_{i=1}^{k} T_{coll_i}}{k} \tag{3.10}$$

$$\tilde{T_{adet}} = \frac{\sum_{i=1}^{k} T_{det_i}}{k} \quad (3.11)$$

$$\tilde{T_{acd}} = \frac{\sum_{i=1}^{k} T_{cd_i}}{k} \quad (3.12)$$

Де k – кількість усіх деталізованих записів.

Після цього за основу коефіцієнта вагомості m=1 береться джерело з найбільшим значенням затримки, а далі за виразом різниці затримок розраховуємо решту m.

Таблиця 3.1 – розрахунок коефіцієнта вагомості m.

| | m | Затримка, години |
|---|---|---|
| Джерело1 | $\tilde{T_{acoll3}} / \tilde{T_{acoll1}}$ | $\tilde{T_{acoll1}}$ |
| Джерело2 | $\tilde{T_{acoll3}} / \tilde{T_{acoll2}}$ | $\tilde{T_{acoll2}}$ |
| Джерело3 | 1 | $\tilde{T_{acoll3}}$ |

В кінці розраховуємо середнє значення на основі вагомого коефіцієнту.

Середнє значення на основі вагомого коефіцієнту $T_{coll}$

$$\frac{m_1 \tilde{T_{call1}} + m_2 \tilde{T_{call2}} + m_3 \tilde{T_{call3}}}{m_1 + m_2 + m_3} \quad (3.13)$$

Середнє значення на основі вагомого коефіцієнту $T_{det}$

$$\frac{m_1 \tilde{T_{det1}} + m_2 \tilde{T_{det2}} + m_3 \tilde{T_{det3}}}{m_1 + m_2 + m_3} \quad (3.14)$$

Середнє значення на основі вагомого коефіцієнту $T_{cd}$

$$\frac{m_1 \tilde{T_{cd1}} + m_2 \tilde{T_{cd2}} + m_3 \tilde{T_{cd3}}}{m_1 + m_2 + m_3} \quad (3.15)$$

## 3.2 Схема тестового середовища та опис його складових

Тестове середовище було побудоване за допомогою використання наступних технологій:

- Cisco NFV – віртуалізація мережевих функцій на базі VMware vSphere ESXi. Є одним з найбільш популярним середовищем, що підтримує найбільший обсяг продуктів від виробників інфокомунікаційних рішень [80].
- Ericsson vEPC – базова мережа LTE від компанії Ericsson. Завдяки віртуалізації та використанню контейнеризації дозволяє масштабування та керування ресурсами з повною підтримкою сервісів Cisco NFV.
- Rhino VoLTE TAS VM – рішення для надання послуг VoLTE від компанії Metaswitch [40, 41]. Гнучка платформа на базі Rhino TAS (Telecom application server), що має відкритий Rhino SDK (Software development kit) для розробки або редагування існуючого функціоналу, з підтримкою VMware vSphere ESXi.
- Oracle BRM (Billing and Revenue Management) - система розрахунку наданих послуг для абонентів в залежності від типу, кількості годин, об’єму трафіку від компанії Oracle [103]. Дана система надає вбудований функціонал створення TAP3, NRTRDE файлів.
- FMS на базі Oracle RDBMS – система розпізнання шахрайства на базі SQL Oracle RDBMS. Використовує різні правила, списки для аналізу завантажених даних у таблиці. Завдяки використання Oracle RDBMS, має вбудований функціонал обробки TAP3, NRTRDE файлів.

•ETL (Extract Transform Load) server – віртуальна машина, що здійснює форматування CDR файлів Rhino та завантаження у базу даних FMS разом з ціною за надання послуг за допомогою Oracle ODI.

•NAS (Network Attached Storage) – виконує функції розподіленої файлової системи через яку FMS отримує доступ до файлів для обробки та ETL сервер виконує функцію насичення файлів CDR Rhino. З'єднання встановлюється за допомогою NFS (Network File System).

• Сімбокс+софтфон використовуються для створення навантаження, наближеного до навантаження на реальній мережі, у тестовому середовищі.

Для тестування методики виявлення за допомогою комплексного використання CDR було надане наступне тестове середовище [39,106], яке складається з наступних елементів (рис. 3.2):

•MME (Mobile management entity) - вузол для обробки сигналізації, переважно пов'язаної з управлінням мобільністю абонентів в мережі EPC.

•SGW (Serving GW) - обслуговуючий шлюз мережі стандарту LTE. SGW є шлюзом між CN (Core Network) і мережею доступу.

•PGW (Packet data network GW) - шлюз до інших мереж передачі даних для мережі LTE. В нашому випадку до IMS

•HSS (Home Subscriber Server) - база даних, призначена для зберігання даних про абонентів.

•PCRF (Policy and Charging Rules Function) - вузол, що відповідає за управління нарахуванням плати за надані послуги зв'язку, а також за якість з'єднань відповідно до заданих конкретному абоненту характеристиками

•OCS/OFCS (Online Charging System/ Offline Charging System – вузол розрахунку наданих послуг для абонентів в залежності від типу, кількості годин, об'єму трафіку.

•P-CSCF (Proxy- Call session control function) – перша точки комутації мобільного користувача з IMS. Надає зв'язок абонента між S-CSCF та забезпечує безпеку сигнальних повідомлень за допомогою захисту інтеграції та шифрування.

•I/S-CSCF (Interrogating/Serving Call session control function) – S-CSCF контролює мобільний пристрій користувача, надає послуги голосових дзвінків (в даному сценарії). I-CSCF забезпечує функцію розподілу та комунікацію з HSS.

•AS (Application server) – вузол, що відповідає за надання послуг VoLTE

•TrGW (Transition Gateway) - вузол для передачі медіа даних IMS до іншої IMS мережі.

•IBCF (Interconnect Border Control Function) – вузол для передачі даних сигналізації IMS до іншої IMS мережі.

•Firewall - віртуальна машина, яка виконує функції фаєрвола.

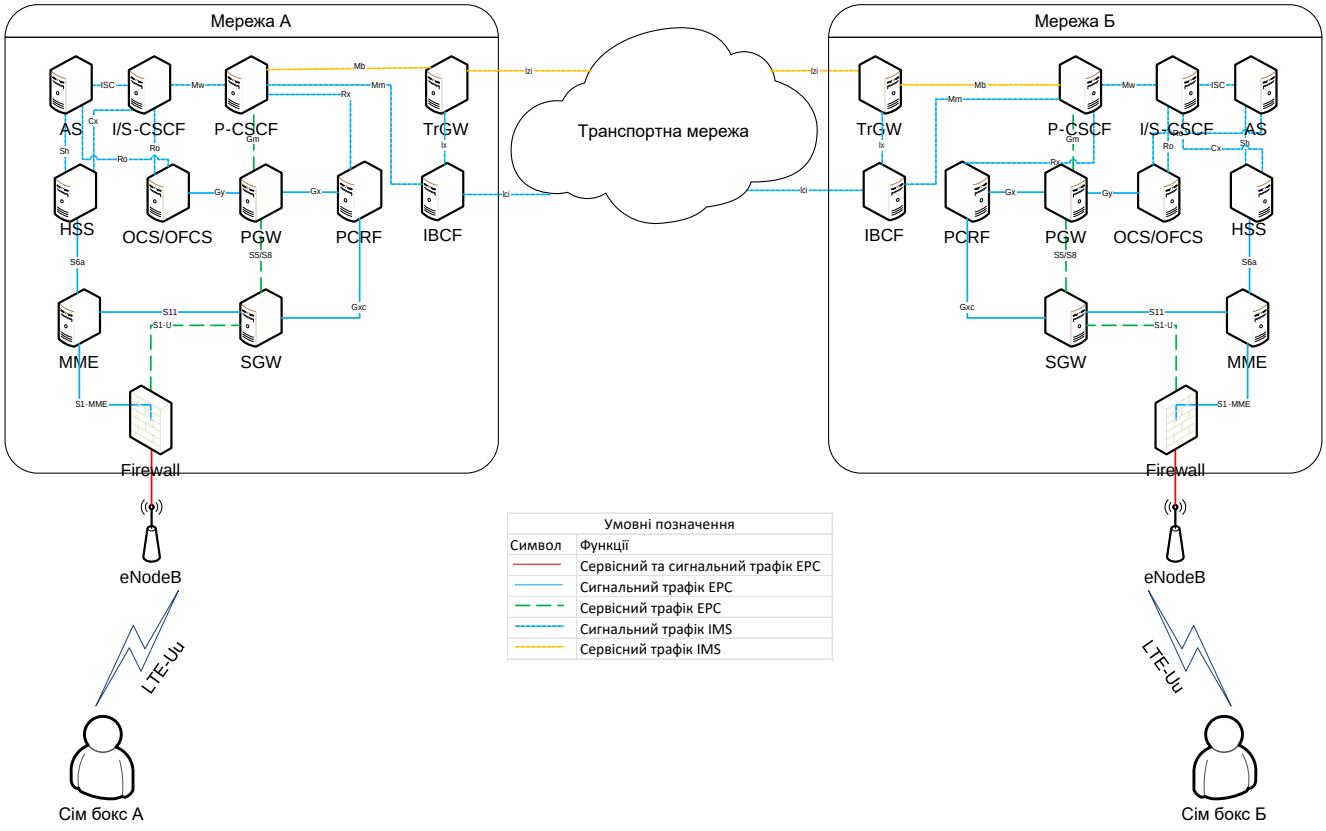
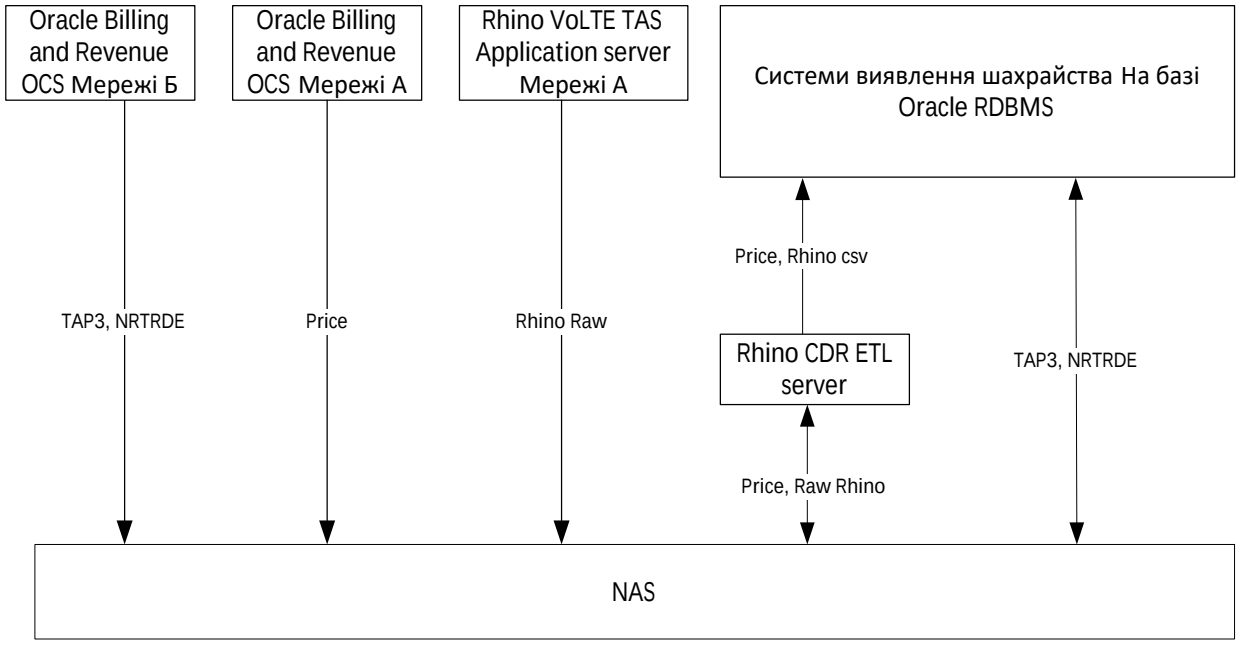



Рисунок 3.2 – Схема тестової мережі

Для обробки даних з тестового середовища була побудована наступна схема взаємодії системи виявлення шахрайства з системами розрахунку абонентів мереж А та Б, AS мережі А та ETL сервера (ETL server), що виконує функції

підготовки даних з AS для їх наступного завантаження до бази даних системи виявлення шахрайства для їх обробки. Для обробки даних NRTRDE, TAP3 система має вбудований механізм, що складається з ASN1 декодеру та парсеру.




| | Файл | Шлях | Комментарі |
|---|---|---|---|
| 1 | NRTRDE | /fraud_detection/NRTRDE/in | Файл ASN1 формату |
| 2 | TAP3 | /fraud_detection/TAP3/in | Файл ASN1 формату |
| 3 | Rhino Raw | /fraud_detection/Rhino/raw | txt файл формату AVP |
| 4 | Price | /fraud_detection/Price/in | csv ціновий файл |
| 5 | Rhino csv | /fraud_detection/Rhino/in | csv файл |

Рисунок 3.3 – високо-рівнева схема підготовки даних

Схему можливо розділити на 2 потоки даних:

1. Потік NRTRDE та TAP3, що безпосередньо передається за допомогою NFS (Network File Sharing) у систему для їх обробки.

2. Потік Price та Rhino CDR, що за допомогою NFS надходять на ETL сервер, де проходять трансформацію у необхідний формат та завантажуються у базу даних системи виявлення шахрайства за допомогою Oracle ODI, який встановлений на ETL сервері.

### 3.2.1 Опис розробленого інтерфейсу для потоку Price та Rhino CDR

CDR з AS потребують додаткової конфігурації, а саме ввімкнення функції створення CDR сесій, що налаштовуються при зміні параметрів у файлі конфігурації sentinel-volte-gsm-config.yaml [41]. У додатку 1 наведено які саме параметри потрібно змінити. По замовчуванню, CDR генеруються у бінарному вигляді, тому платформа також надає SDK та інструмент (list-cdrs.sh) [76,78], що дозволяють трансформувати файл у форму, яку може читати людина. Для цього був написаний bash код binary_to_human_read.sh (додаток 2), що періодично (кожні 5 хвилин) запускає інструмент list-cdrs.sh за допомогою cron (додаток 3) та виконує функцію переміщання файлів до NAS за допомогою NFS. Зазвичай такий CDR має назву «cdr_101_*.txt», де замість зірочки вказується час створення файлу у форматі часу UNIX.

NFS (Network File Sharing) – це протокол, що дозволяє надавати доступ до папок та файлів різним Linux клієнтам. Зазвичай ці файли та папки зберігаються на файловому сервері NAS. Даний функціонал дозволяє робити оновлення та обробку файлів без його завантаження на сам сервер.

Файл з цінами на наданий сервіс (export_product) теж експортується з OCS за допомогою вбудованого інструментарію Oracle [103]. Для цієї функції теж був написаний bash (додаток 4) та налаштований cron з інтервалами роботи 5 хвилин (додаток 5).

Для наступного завантаження та обробки CDR у базі даних Oracle, необхідно його трансформувати у формат, який сприймається реляційними базами даних, бо цей файл має структуру AVP, що характерна для нереляційних баз даних та їх сховищ. До форматів, які підтримують реляційні бази належать csv, tsv, xml, excel.

Rhino CDR складається з двох частин - Header, що надає загальну інформацію про платформу з якою надійшов CDR та AvpCdr (AVP - Attribute Value Pair), що надає інформацію про надані послуги, ідентифікатори сесій, абонентів та інше [42,72-75,77]. Складові Header та AvpCdr наведені у таблицях 3.2 та 3.3.

Таблиця 3.2 – складові Header Rhino CDR

| **Колонтитул** | **Складова** | **Опис** | **Приклад даних** |
|---|---|---|---|
| ra_name | - | Назва інструменту | CDR Generation |
| ra_vendor | - | Назва виробника | OpenCloud |
| ra_version | - | Версія інструменту | 2.3 |
| ra_release | - | Вихідна версія | 2.3.0.0-M1 |
| ra_build | - | Дата створення версії | 20160706024526 |
| ra_revision | - | Ревізія | cdr-ra/2.3.0@d55f6a1 |
| description | - | Опис операції | CDR session |
| rhino_node | - | Номер вузла у мережі | 101 |
| ra_entity | - | Тип даних | cdr |
| hostname | - | Назва серверу на якому знаходяться CDR | mortadell |

Таблиця 3.3 – складові AvpCdr Rhino CDR

| **Колонтитул** | **Складова** | **Елемент** | **Опис** | **Приклад даних** |
|---|---|---|---|---|
| Subscription-Id | Subscription-Id-Type | | Код типу наданого сервісу | 2=SipCall |
| | Subscription-Id-Data | | Телефонний номер користувача | 34600000002 |
| IMS-Information | Event-Type | [SIP-Method]<br>[ Event ]<br>[ Expires ] | Тип інфокомунікаційних послуг або подій за які нараховується плата на основі | |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | повідомлення ACR та (або) CCR | |
| | Role-Of-Node | | Надає інформацію стосовно функції яку надає вузол під час сеансу | 0 ORIGINATING_ROLE Вузол обслуговує абонента, що виконує дзвінок 1 TERMINATING_ROLE Вузол обслуговує абонента, якому дзвонять 2 FORWARDING_ROLE Вузол виконує перенаправлення |
| | Node-Functionality | | Містить код, що визначає функцію вузла | 0 S-CSCF 1 P-CSCF 2 I-CSCF 3 MRFC 4 MGCF 5 BGCF 6 AS 7 IBCF |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | | 8 S-GW<br>9 P-GW<br>10 HSGW<br>11 E-CSCF<br>12 MME<br>13 TRF<br>14 TF<br>15 ATCF<br>16 Proxy Function<br>17 ePDG<br>18 TDF |
| | User-Session-Id | | Ідентифікатор сесії у форматі UTF8String | cXD7xzRmA1w4iWBzT9XMdQ |
| | Session-Priority | | Пріоритет сесії. Має постійне значення 2 | PRIORITY_2(2) |
| | Calling-Party-Address | | Номер абоненту, що виконує дзвінок у форматі UTF8String | 34600000002 |
| | Called-Party-Address | | Номер абоненту, що отримує | 34600000003 |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | дзвінок у форматі UTF8String | |
| | Called-Asserted-Identity | | Поле типу UTF8String, що має у собі значення адреси фінального абонента, що отримує дзвінок | sip:conf-factory@localhost:5060 |
| | Requested-Party-Address | | Поле типу UTF8String, що має у собі значення адреси групи, на яку сесія була надіслана спочатку. Поле присутнє у випадку, коли значення відрізняється від Called-Party-Address | sip:conf-factory@localhost:5060 |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | Inter-Operator-Identifier | Originating-IOI | Код мережі у форматі UTF8String, з якої надходить сесія | bea.net |
| | | Terminating-IOI | Код мережі у форматі UTF8String, яка приймає сесію | bea.net |
| | IMS-Charging-Identifier | | Поле формату UTF8String, що зберігає значення ICID (IMS Charging Identifier), яке генерується IMS вузлом для SIP сесії | |
| | SDP-Session-Description | | Поле формату UTF8String, що зберігає складову строки SDP | [v=0], [o=- 1000 1000 IN IP4 127.0.0.1], [s=test-session], [t=0 0] |
| | SDP-Media-Component | [SDP-Media-Name] [SDP-Media- | Це групове значення, що містить | |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | Description] [Local-GW-Inserted-Indication] [IP-Realm-Default-Indication] [Transcoder-Inserted-Indication] [Media-Initiator-Flag] [Media-Initiator-Party] [3GPP-Charging-Id] [Access-Network-Charging-Identifier-Value] [SDP-Type] | інформацію про медіа, що використовується під час сесії IMS | |
| | Cause-Code | | Поле формату Integer32, що заповнюються кодовим значенням з вузла IMS. Це | Значення ≥ 1 надають інформацію про неуспішні причини, а значення ≤ 0 – про |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | значення надає причину кінця сесії. | успішні. 0 нормальне закінчення сесії |
| | Access-Network-Information | | Поле формату OctetString, що містить одну строчку з заголовку P-Access-Network-Info | |
| | Early-Media-Description | [SDP-TimeStamps ] [SDP-Media-Component] [SDP-Session-Description] | Це група, що описує SDP сесію, медіа параметри та часи для встановлення кожного медіа компоненту | SDP-TimeStamps[ SDP-Offer-Timestamp[1467865834000], SDP-Answer-Timestamp[1467865834000]], SDP-Session-Description[v=0], SDP-Session-Description[o=- 1000 1000 IN IP4 127.0.0.1], SDP-Session-Description[s=test-session], SDP-Session-Description[t=0 0] |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | IMS-Communication-Service-Identifier | | Поле формату UTF8String, що містить у собі ICSI (IMS Communication Service Identifier), що міститься у е P-Asserted-Service заголовку SIP запиту | |
| User-Equipment-Info | {User-Equipment-Info-Type } {User-Equipment-Info-Value } | | Поле заповнюється інформацією з Ro інтерфейсу | IMEI AA-BBBBBB-CCCCCC-D |
| Multiple-Services-Credit-Control | Service-Identifier | | Поле містить Id використаної послуги | 50 |
| OC-Call-Type | | | Тип дзвінка до абонента | MOC(1) – дзвінок був ініційовано абонентом MOC_3RDPTY(2) - дзвінок був ініційовано через |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | | (наприклад, через HTTP)<br>MTC(3) – абоненту подзвонили<br>MFC(4) – дзвінок з перенаправленням<br>EMERGENCY_CALL(9) – дзвінок був розпізнаний як аварійний |
| OC-Service-Type | | | Містить значення повідомлення з якої почалася сесія | UNKNOWN(1) – причина початку сесії невідома.<br>SipCall(2) – сесія почалася після отримання SIP INVITE<br>Subscription(3) – сесія почалася після отримання SIP SUBSCRIBE<br>Message(5) – сесія почалася після отримання SIP MESSAGE |
| OC-Charging-Result | | | Містить значення код результату | Якщо Ro сесія не використовувалася, то поле приймає |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | Diameter CCA. | значення -1 |
| OC-OCS-Session-Id | | | Id сеансу під час комунікації з OCS | diameterclient;diameterro-0;1529370976;0;08tE5q0fRdQRrus7F4FWqg |
| OC-OCS-Session-Termination-Cause | | | Надає причину закінчення сеансу по інтерфейсу Ro | NORMAL_SESSION_COMPLETION = 0<br>ERROR_CCA = 1<br>CREDIT_LIMIT_REACHED = 2<br>OCS_ABORT = 3<br>TCC_EXPIRED = 4<br>CREDIT_CONTROL_FAILURE = 5<br>CLIENT_ABORT = 6 |
| OC-Sentinel-Error | | | Поле помилок платформи | None = 1<br>OcsTimeout = 2<br>OcsCommunicationFailure = 3<br>SentinelOverload = 4<br>ProtocolError = 5<br>InternalError = 6<br>MappingError = 7<br>OtherError = 8 |

| **Колонтитул** | **Складова** | **Елемент** | **Опис** | **Приклад даних** |
|---|---|---|---|---|
| OC-Charging-Instance | OC-Charging-Instance-Name | | Імя OCS, що використовувався у даному сеансі | scur_charging_instance |
| | OC-Session-Counter | OC-Session-Counter-Address | Група у форматі UTF8String, що складається з [OC-Session-Counter-Address-Key] та [OC-Session-Counter-Address-Value], яка уявляє собою пару для ідентифікації лічильника для розрахунку оплати | OC-Session-Counter-Address-Key[Subscriber-Id], OC-Session-Counter-Address-Value[tel:346000000002]; OC-Session-Counter-Address-Key[Service-Id], OC-Session-Counter-Address-Value[1]; OC-Session-Counter-Address-Key[Cc-Unit-Type], OC-Session-Counter-Address-Value[Cc-Time] |
| | | OC-Cumulative-Committed-Used | Поле формату Integer64, це сума Used-Service-Units у | 1000 |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | CCR Multiple-Service-CreditControl | |
| | | OC-Cumulative-Granted | Поле формату Integer64, це сума Granted-Service-Units у CCR Multiple-Service-CreditControl | 60000 |
| | | OC-Cumulative-Granted-Refund | Поле формату Integer64, це сума Requested-Service-Units у CCR(REFUND) Multiple-Service-CreditControl, які отримали успішно CCA відповідь | 0 |
| | | OC-Cumulative-Requested[], | Поле формату Integer64, це сума Requested-Service-Units у CCR Multiple- | 60000 |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | Service-CreditControl | |
| | | OC-Cumulative-Requested-Refund[], | Поле формату Integer64, це сума Requested-Service-Units у CCR(REFUND) Multiple-Service-CreditControl | 0 |
| | | OC-Cumulative-Sent-Used[] | Поле формату Integer64, це сума Used-Service-Units у CCR Multiple-Service-CreditControl | 1000 |
| | | OC-Cumulative-Suspended-Duration | Кумулятивна тривалість усіх періодів, де була затримка нарахування за сеанс у мілісекундах | 0 |
| | | OC-Reported-Used | Одиниці відзвітовано | 0 |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | як використані, які будуть відправлені у наступному CCR | |
| | | OC-Pending-Requested | Одиниці, що будуть запитані у наступному CCR | 0 |
| | | OC-Start-Time | Час початку голосового дзвінку | 1467865834000 |
| | | OC-End-Time | Час кінця голосового дзвінку | 1467865836000 |
| OC-Event-Id | | | Поле заповнюється якщо запит на сеанс був SIP SUBSCRIBE, SIP NOTIFY, SIP REFER | Для SIP SUBSCRIBE, SIP NOTIFY Event: Для SIP REFER - CSeq |
| OC-Call-Id | | | Call-ID сеансу SIP | 08tE5q0fRdQRrus7F4FWqg |
| OC-End-Session- | | | Поле надає інформацію | |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| Cause | | | про причину закінчення сесії. | |
| OC-Session-Start-Time | | | Поле надає інформацію про час, коли запит Initial Request був оброблений | 1467865836000 |
| OC-Session-Established-Time | | | Поле надає час, коли на сесію була відповідь, тобто повідомлення ACK на повідомлення 2xx-INVITE | 1467865836000 |
| OC-Session-End-Time | | | Поле надає час, коли сесія була закінчена | 1467865836000 |
| OC-OCS-Destination-Realm | | | Diameter realm платформи OCS під час сеансу | |
| User-Name | | | Використовує Private Id з | Private Id |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | даних реєстрації | |
| OC-Selection-Key | | | | Поле для майбутньої розробки |
| OC-Play-Announcement-Id | | | ID announcement, що використовується піж час сесії, якщо є в наявності | |
| OC-Terminating-Domain | | | Дане поле заповнюється у випадку, якщо поле є в сигналізації SIP іншої платформи | CS<br>PS=EUTRAN<br>PS=NR<br>PS=WLAN |
| OC-Billing-ID | | | BillingID заповнюється з запиту ReOriginated CDMA | Поле стосується лише CDMA |
| OC-Access-Network-MCC-MNC | OC-MCC-MNC | | Поле типу UTF8String, яке тримає значення | |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | мережі з якої надано доступ. Цю інформацію отримує з заголовку P-Access-Network-Info | |
| | OC-Age-Of-Information | | Час реєстрації у мережі | |
| OC-Visited-Network-MCC-MNC | OC-MCC-MNC | | Поле типу UTF8String, яке тримає значення MNC, MNC мережі з якої надано доступ. Цю інформацію отримує з заголовку P-Visited-Network-Id | |
| | OC-Age-Of-Information | | Час реєстрації у мережі | |
| OC-IMSI-MCC-MNC | OC-MCC-MNC | | Поле типу UTF8String, яке тримає | |

| Колонтитул | Складова | Елемент | Опис | Приклад даних |
|---|---|---|---|---|
| | | | MNC, MNC мережі з якої надано доступ. Цю інформацію отримує з IMSI під час реєстрації або SRI response | |
| | OC-Age-Of-Information | | Час реєстрації у мережі або SRI response | |

Як можна побачити, Rhino CDR не надає інформації стосовно мінімальної ціни за сервіс, ні валюти для розрахунку послуг. Тому, для того, щоб отримати рахунок за надання послуг, наступні AVP можуть бути використані для визначення мережі OC-Access-Network-MCC-MNC, OC-Visited-Network-MCC-MNC, OC-IMSI-MCC-MNC, Inter-Operator-Identifier (Originating-IOI, Terminating-IOI). Перші 3 AVP беруть інформацію з P-Asserted-Service, коли абонент що робить виклик, надсилає запит до викликаємого абоненту, тому ці поля надають інформацію лише про одного абонента. А от поле Inter-Operator-Identifier надає інформацію про кінцеві мережі, що надають послуги користувачам.

Файл export_product.csv складається з наступних полів, що наведені у таблиці 3.4.

Таблиця 3.4 – складові export_product.csv

| Назва рядка | Опис | Приклад даних |
|---|---|---|
| Service_name | Назва послуги, що використовує користувач | SipCall |

| **Назва рядка** | **Опис** | **Приклад даних** |
|---|---|---|
| Service_type | Направлення послуги – отримувач або відправник | MOC |
| Network_TADIG | Код мережі на яку користувач робить дзвінки | USAHI<br>SWE01 |
| MCC | Mobile country code | 255 |
| MNC | Mobile network code | 01 |
| IOI | Inter-Operator-Identifier | bea.net |
| Min_price | Ціна за надання послуг | 10 |
| Currency | Валюта, у якій надаються послуги | EUR |

Для визначення типу наданих послуг використаємо поля OC-Service-Type, OC-Call-Type. А для розрахунку тривалості сесії використаємо поля OC-Start-Time, OC-End-Time. Фінальний запис CDR та взаємозв’язок між полями у форматованому файлі Rhino CDR та export_product.csv наведено у таблиці 3.4. Ця таблиця має назву fraud.ur_ims_records. Таблиця була підготовлена на основі структури існуючих таблиць для TAP3 та NRTRDE файлів з розрахунком IMS сервісів.

Таблиця 3.5 – взаємозв’язок між полями fraud.ur_ims_records та вхідними даними.

| **Назва рядка** | **Тип поля** | **Рядок RhinoAVR** | **Рядок export_product** |
|---|---|---|---|
| cdr_id | Number(16), cdr_id_seq.nextval | - | - |
| Loading_time | Date, systime | | - |
| source_name | Varchar2(30 char) | Header { hostname=mortadella } | |
| file_name | Varchar2(30 char) | Назва файлу Rhino | |

| **Назва рядка** | **Тип поля** | **Рядок RhinoAVR** | **Рядок export_product** |
|---|---|---|---|
| | | CDR | |
| Service_name | Varchar2(10 char) | AvpCdr {avps=[ OC-Service-Type(Ext,Ext)[] ]} | |
| Service_type | Varchar2(10 char) | AvpCdr { avps=[ OC-Call-Type(Ext,Ext)[]]} | |
| session_id | Varchar2(65 char) | AvpCdr { avps=[ User-Session-Id[] }] | |
| Originated_network | Varchar2(30 char) | Originating-IOI | |
| Destination_network | Varchar2(30 char) | Terminating-IOI | |
| Currency | Varchar2(8 char) | | Currency |
| Min_price | Number(5) | | Min_price |
| Price_sdr | Number(20,6) | Duration*Min_price | |
| Start_call_time | Date | AvpCdr { avps=[ OC-Start-Time[] }] | |
| Session_start_time | Date | OC-Session-Start-Time | |
| Session_establishment_time | Date | OC-Session-Established-Time | |
| End_call_time | Date | AvpCdr { avps=[ OC-End-Time[] }] | |
| Session_end_time | Date | OC-Session-End-Time | |
| Session_duration | Number(20) | (AvpCdr { avps=[ OC-End-Time[] }] - AvpCdr { avps=[ | |

| **Назва рядка** | **Тип поля** | **Рядок RhinoAVR** | **Рядок export_product** |
|---|---|---|---|
| | | OC-Start-Time[] }]) /1000 | |
| UTC_OFFSET | Varchar2(5 char) | UTC+0 | |
| Served_msisdn | Varchar2(32 char) | Subscription-Id-Data | |
| Originated_number | Varchar2(32 char) | Calling-Party-Address | |
| Originated_imei | Varchar2(32 char) | If AvpCdr { avps=[ OC-Call-Type(Ext,Ext)[MOC] }], then User-Equipment-Info-Value, Else ‘Empty’ | |
| Destination_number | Varchar2(32 char) | Called-Party-Address | |
| Destination_imei | Varchar2(32 char) | If AvpCdr { avps=[ OC-Call-Type(Ext,Ext)[MTC] }], then User-Equipment-Info-Value, Else ‘Empty’ | |
| Id_third_party_num | Varchar2(32 char) | Called-Asserted-Identity | |
| Id_dialed_number | Varchar2(32 char) | Requested-Party-Address | |
| Nt_charge_id | Varchar2(20 char) | OC-Call-Id | |
| Serving_Node_Role | Varchar2(20 char) | Role-Of-Node | |

| **Назва рядка** | **Тип поля** | **Рядок RhinoAVR** | **Рядок export_product** |
|---|---|---|---|
| Serving_Node_Function | Varchar2(20 char) | Node-Functionality | |
| Charging_Instance | Varchar2(40 char) | OC-Charging-Instance-Name | |
| Nt_termination_cause | Number(5) | OC-OCS-Session-Termination-Cause | |
| Session_ICSI | Varchar2(40 char) | IMS-Communication-Service-Identifier | |
| Session_ICI | Varchar2(40 char) | IMS-Charging-Identifier | |

Код bash наведений у додатку 6 виконує функцію витягування необхідних полів, наведених у таблиці 3.5, з Rhino AVP CDR та проводить форматування у формат csv. Якщо по якийсь причині поле надійшло порожнім, код залишає його пустим без зміни розташування рядків у csv файлі cdr_101_*.txt.csv.

Для завантаження та створення записів CDR з розрахунком ціни за послугу, була створена наступна схема в Oracle ODI [70], яка наведена на рисунку 3.4.

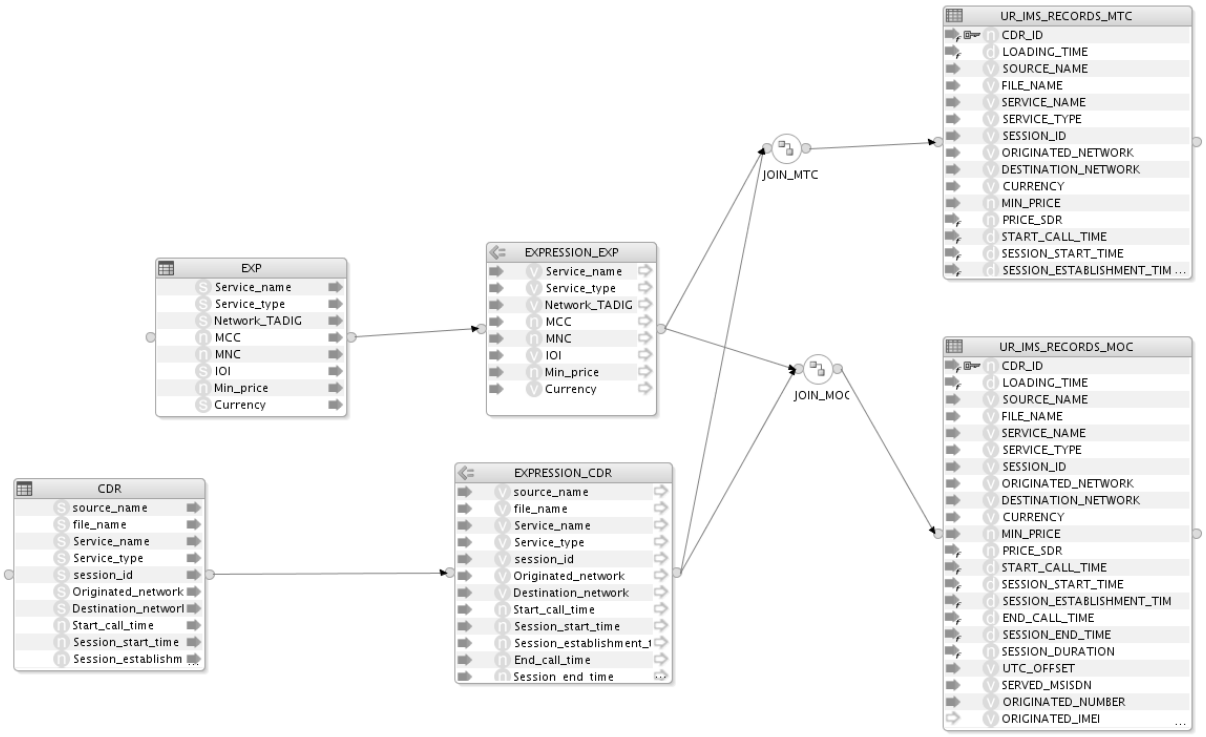



Рисунок 3.4 – логічна схема взаємодії файлів export_product.csv та cdr_101_*.txt.csv

Схема працює наступним чином:

1. Спочатку система завантажує файли з робочих папок «/media/sf_fraud_detection/Price/in» та «/media/sf_fraud_detection/Rhino/in» у тимчасові таблиці C$_0EXP, C$_1CDR.

2. Після цього для операції об’єднання даних (JOIN_MOC, JOIN_MTC) значення з тимчасових таблиць перетворюються у вирази (EXPRESSION_EXP, EXPRESSION_CDR).

*3.* Далі таблиці об’єднуються за виразами для JOIN_MTC *CDR.Service_name=EXP.Service_name and EXP.Service_type=CDR.Service_type and CDR.Destination_network=EXP.IOI* та JOIN_MOC *CDR.Service_name=EXP.Service_name and EXP.Service_type=CDR.Service_type and CDR.Originated_network=EXP.IOI*

4. Після об’єднання дані завантажуються у таблицю fraud.ur_ims_records, де часові інтервали сесії проходять трансформацію з часових інтервалів ОС Unix за допомогою виразу *(to_date('1970-01-01 00:00:00','YYYY-*

*MM-DD HH24:MI:SS') + numtodsinterval((CDR.Start_call_time/1000),'SECOND'))*, розраховується ціна за надання послуг *(EXP.Min_price*(CDR.End_call_time-CDR.Start_call_time)/60000)* та тривалість сесії *(CDR.End_call_time-CDR.Start_call_time)/60000*

5. Після завантаження запису в таблицю, система періодично перевіряє її та завантажує нові CDR у пам'ять додатка, де вони проходять обробку.

На фізичному рівні схеми взаємодії для завантаження інформації з файлів у тимчасові таблиці використовуються Loading Knowledge Module (LKM) SQL to SQL (Built-in) Global (рис. 3.5). Це вбудований модуль ODI для взаємодії реляційної бази даних Oracle з файлами даних для завантаження [69].

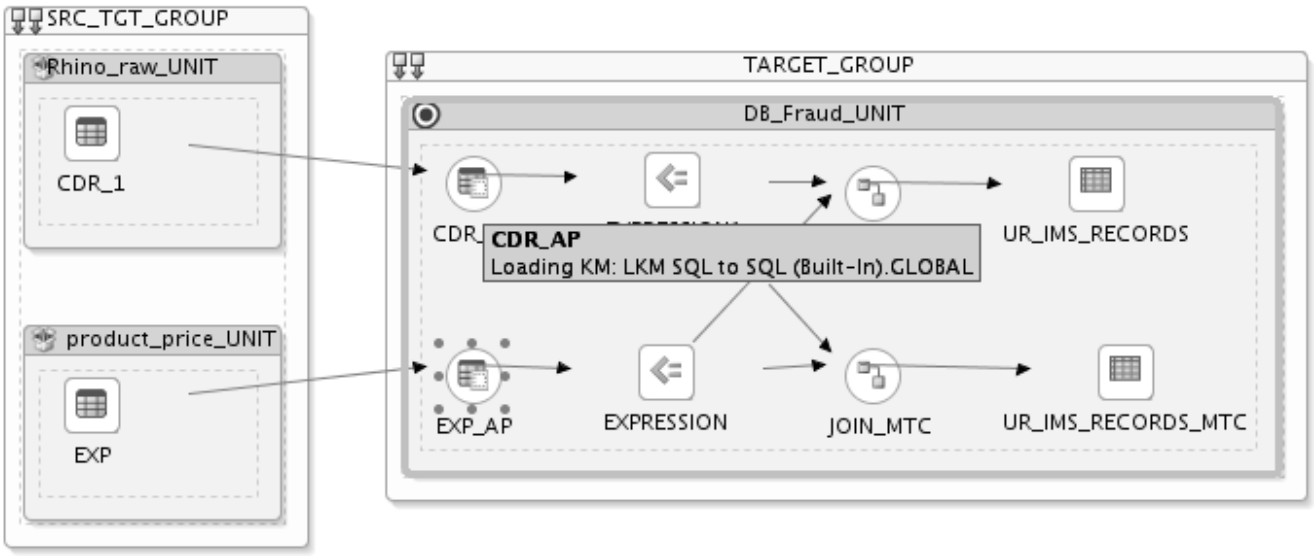



Рисунок 3.5 – фізична схема взаємодії файлів export_product.csv та cdr_101_*.txt.csv

Після того, як ODI завантажив дані у таблиці, CDR файли, що знаходилися у папках «/media/sf_fraud_detection/Rhino/in» переміщуються у «/media/sf_fraud_detection/Rhino/done». Завдяки вбудованому менеджеру сценаріїв (Scenarios) дана схема відпрацьовує з періодом 5 хвилин.

У додаток 8 наведено xml файл схеми з необхідними параметрами, що можуть бути завантажені до іншої системи ODI для відтворення схеми.

Розроблений алгоритм має наступні переваги у порівнянні з мережним зондом:

- Відсутність залежності від місцезнаходження обладнання базової мережі;
- Відсутність залежності від спеціалізованого обладнання;
- Усунута залежність витрат від змін у інфраструктурі базовій мережі.

Тому формула розрахунку оперативних витрат набуває наступного вигляду

$$Cost_{operation} = M_{total} * S_{avg} + \sigma_l + \sigma_{swl} \quad (3.15)$$

Де $\sigma_{swl}$ ціна ліцензії ODI.

Якщо розрахувати операційні місячні витрати на мережний зонд та алгоритм, то виявиться, що розроблений алгоритм у 13 разів дешевше. Результати розрахунків наведені у таблиці 3.6

Таблиця 3.6 – операційні витрати мережного зонду та алгоритму Rhino.

| | Мережний зонд | Алгоритм Rhino | Різниця |
|---|---|---|---|
| $M_{total} * S_{avg}$ | 600 | 300 | |
| $IT_{depi}$ | 825,25 | - | |
| $\sigma_l$ | 0 | 208,3333 | |
| $\sigma_{swl}$ | 6509,667 | 110 | |
| $IT_{spli}$ | 109,8333 | 0 | |
| Операційні витрати $, місяць | 8044,75 | 618,33 | 13 |

## 3.3 Розрахунок ефективності розпізнання шахрайства

### 3.3.1 Загальна обробка CDR всередині системи виявлення шахрайства

Після завантаження даних до таблиць ur_ims_records за допомогою JVM (Java virtual machine), система починає обробку даних, що проходять наступний шлях між таблицями fraud.ur_ims_records->fraud.pos_prealerts->total_alerted_cdrs/ fraud.ims_alerted_cdrs->fraud.alerts->fraud.cases. Ці таблиці виконують наступні функції:

fraud.ur_ims_records – таблиця, що було розроблена під час дослідження для CDR IMS Rhino;

fraud.pos_prealerts – таблиця, що використовується JVM для створення тимчасових сповіщень про шахрайство. Ця таблиця зберігає усі типи сповіщень, які стали повноцінними сповіщеннями або система їх відкинула, бо сприйняла їх за дублікат.

fraud.ims_alerted_cdrs – це таблиця, яка зберігає CDR, на основі яких було створення оповіщення про шахрайство. За замовчуванням, записи у total_alerted_cdrs дублюють інформацію у fraud.ur_ims_records, але ще надають час занесення у таблицю після створення оповіщення, ідентифікатор оповіщення. Таблиця total_alerted_cdrs зберігає інформацію про усі CDR на основі яких було створення оповіщення незалежно від типу джерела (тобто вона об'єднує CDR усіх типів джерел даних).

fraud.alerts – це таблиця, яка зберігає інформацію про створення оповіщення. Ця таблиця є загальною для усіх типів джерел даних та надає інформацію про час створення оповіщення, кількість записів CDR, ідентифікатор правила на основі якого було створено оповіщення, номер підозрілого абонента.

fraud.cases – це таблиця, яка зберігає інформацію про створені справи шахрайства. Після створення оповіщення, система або додає його до існуючої справи або створює нову на суб'єкта оповіщення, якщо такої не існує або вона була закрита.

Для NRTRDE та TAP3 шлях обробки виглядає так само fraud.ur_nrtrde_records->fraud.pos_prealerts->total_alerted_cdrs/ fraud.nrtrde_alerted_cdrs->fraud.alerts->fraud.cases та fraud.ur_tap3_records->fraud.pos_prealerts->total_alerted_cdrs/fraud.tap3_alerted_cdrs->fraud.alerts->fraud.cases. Але до цього процесу, спочатку файли проходять декодування за допомогою вбудованого процесу ASN1_decoder на віртуальній машині системи виявлення шахрайства, форматуються у csv формат та завантажуються у таблиці fraud.ur_nrtrde_records та fraud.ur_tap3_records.

Повний цикл обробки у системі на базі Oracle RDBMS наведено на рисунку 3.6.

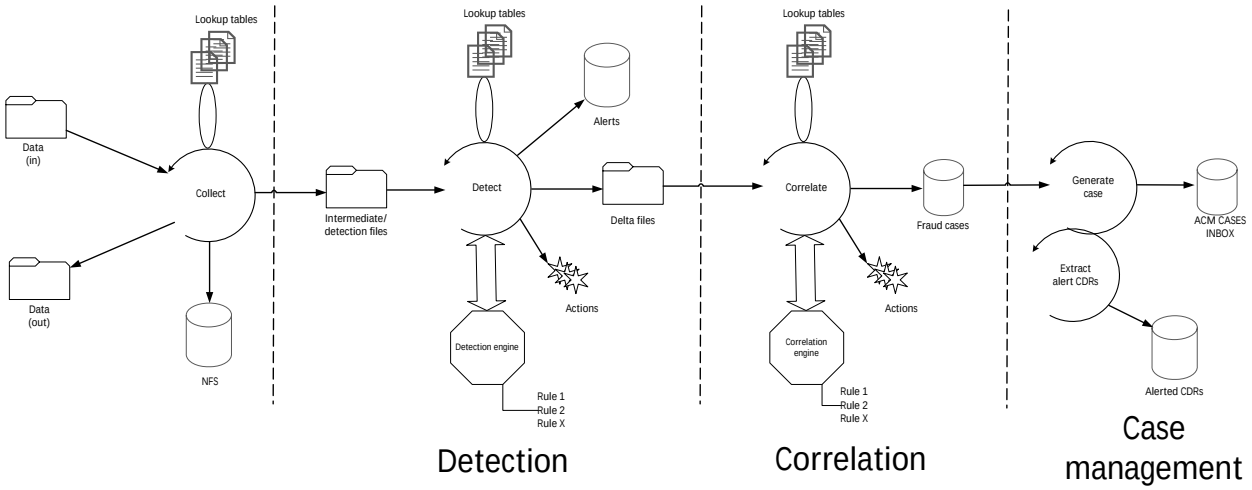



Рисунок 3.6 – Схема обробки трафіку системи виявлення шахрайства.

Як можна помітити, система у кожному етапі використовує lookup таблиці. До цих таблиць належать списки абонентів шахраїв, списки ідентифікаторів мереж, які належать шахраям та таблиці, що дозволяють розпізнати до якого джерела даних належить CDR, яке правило або правила для оброблювати того чи іншого типу даних.

До lookup таблиць належать fraud.sources, fraud.rule_sources, fraud.rules, fraud.conditions, fraud.hotlist, fraud.hotlist_values.

fraud.sources – містить інформацію про джерела даних та їх унікальні ідентифікатори.

fraud.rule_sources – містить інформацію про належність правил до джерел даних визначених у fraud.sources.

fraud.rules – ця таблиця зберігає в собі інформацію про створені користувачами правила, а саме унікальний ідентифікатор правила, назва умов, що використовуються для визначення підозрілого трафіку та граничні умови до яких належать часовий проміжок моніторингу, кількість необхідних підозрілих сесій для створення справи про шахрайство, тощо.

fraud.conditions – таблиця, що містить інформацію про умови створені користувачами для відстеження того чи іншого типу трафіка. Умови можуть складатися з будь-яких полів таблиць ur_*_records та списки, що використовують умови булі алгебри для об’єднання чи виключення тої чи іншої умови.

fraud.hotlist – містить інформацію про існуючі списки, створені користувачами. Містить таку інформацію як час створення списку, ім’я користувача, назву списку та його унікальний ідентифікатор.

fraud.hotlist_values – це таблиця, що зберігає значення підозрілих суб’єктів або ідентифікатори несумлінних операторів мобільного зв’язку. Кожне значення містить час створення значення та час до якого це значення повинне існувати в списку до якого воно належить.

Для виявлення шахрайства були створені однакові правила до кожного типу джерел. У таблиці 3.7 наведено ці правила.

Таблиця 3.7 – взаємозв’язок між полями fraud.ur_ims_records та вхідними даними

| # | **Визначення трафіку для моніторингу** | **Визначення підозрілої поведінки** | **Граничний коефіцієнт для оповіщення** | **Інтервал моніторингу** | **Підозрюваний суб’єкт** |
|---|---|---|---|---|---|
| 1 | MOC, (terminating network - SWE01 or terminating network – bea.net), calling number is not | duration>=30 min and duration<=1800 min | count more or equals 3 calls based on originated IMSI/MSISDN | 1 hour time frame | Originated MSISDN, Originated IMSI |

| # | **Визначення трафіку для моніторингу** | **Визначення підозрілої поведінки** | **Граничний коефіцієнт для оповіщення** | **Інтервал моніторингу** | **Підозрюваний суб'єкт** |
|---|---|---|---|---|---|
| | whitelist | | | | |
| 2 | MOC, (terminating network - SWE01 or terminating network – bea.net), calling number is not present in whitelist | duration>20 sec | count more or equals 10 calls based on originated IMSI/MSISDN | 2 hours time frame | Originated MSISDN, Originated IMSI |
| 3 | MOC, (terminating network - SWE01 or terminating network – bea.net), calling number is not present in whitelist | duration>20 sec and duration<30 sec | sum duration more or equals 1 hours based on originated IMSI/MSISDN | 6 hours time frame | Originated MSISDN, Originated IMSI |
| 4 | MOC, (terminating | duration>20 sec and | sum duration more or equals | 3 hours time frame | Terminated MSISDN, |

| # | **Визначення трафіку для моніторингу** | **Визначення підозрілої поведінки** | **Граничний коефіцієнт для оповіщення** | **Інтервал моніторингу** | **Підозрюваний суб'єкт** |
|---|---|---|---|---|---|
| | SWE01 or terminating network – bea.net), calling number is present in whitelist | duration<30 sec | on terminated IMSI/MSISDN | | Terminated IMSI |
| 5 | MTC, (originating network - SWE01 or originating network – bea.net), called number is not present in whitelist | duration>=30 min and duration<=1800 min | count more or equals 3 calls based on originated IMSI/MSISDN | 1 hour time frame | Originated MSISDN, Originated IMSI |
| 6 | MTC, (originating network - SWE01 or originating network – bea.net), | duration>20 sec | count more or equals 10 calls based on originated IMSI/MSISDN | 2 hours time frame | Originated MSISDN, Originated IMSI |

| # | **Визначення трафіку для моніторингу** | **Визначення підозрілої поведінки** | **Граничний коефіцієнт для оповіщення** | **Інтервал моніторингу** | **Підозрюваний суб'єкт** |
|---|---|---|---|---|---|
| | is not present in whitelist | | | | |
| 7 | MTC, (originating network - SWE01 or originating network – bea.net), called number is not present in whitelist | duration>20 sec and duration<30 sec | sum duration more or equals 1 hours based on originated IMSI/MSISDN | 6 hours time frame | Originated MSISDN, Originated IMSI |
| 8 | MTC, (originating network - SWE01 or originating network – bea.net), called number is not present in whitelist | duration>20 sec and duration<30 sec | sum duration more or equals 1 hours based on terminated IMSI/MSISDN | 3 hours time frame | Terminated MSISDN, Terminated IMSI |

### 3.3.2 Апробація виявлення шахрайства при комплексному використанні CDR з різних джерел

Для апробації результатів дослідження була підготовлена схема середовища, яке наведено у минулому підрозділі та вміщує в собі 980 абонентів, де 490 віртуальних номерів знаходиться на мережі А, а інші 490 на мережі Б. Дана схема була побудована таким чином, щоб наблизити її поведінку до реальної мережі. Кількість сесій в такій мережі може варіюватися від 7 до 8 тисяч за 24 години. У таблиці 3.8 наведено кількість файлів та записів, що були згенеровані у день розрахунку. Для NRTRDE та TAP3 кількість файлів та записів не є однаковою, бо в реальному середовищі один файл може мітити у собі від 1 до 18 тисяч записів.

Таблиця 3.8 – кількість файлів та записів створених у день апробації

| **Тип даних** | **Кількість файлів** | **Кількість записів у файлах** |
|---|---|---|
| Rhino CDR | 7472 | 7472 |
| NRTRDE | 101 | 6227 |
| TAP3 | 13 | 8605 |

Щоб розрахувати ефективність кожного джерела, було підготовлено 3 SQL запита для кожного типу даних (додатки 9-11). Після того, зібрана інформація оброблюється по методиці описаній на початку цього розділу, а саме розраховується $T_{coll}$ для кожного оповіщення та за допомогою формули ceiling у ексель розбиваються на інтервали по 10 хвилин та визначається середньозважене значення $T_{coll}$. Результати розрахунків наведені у таблиці 3.9.

Таблиця 3.9 – кількість файлів та записів створених у день апробації

| **NRTRDE** | | **TAP3** | |
|---|---|---|---|
| **Часові проміжки** | **Кількість cdr_load_delay_ceiling** | **Часові проміжки** | **Кількість cdr_load_delay_ceilin** |
| 00:10:00 | 4 | 03:30:00 | 2 |
| 00:20:00 | 2 | 11:50:00 | 2 |
| 00:30:00 | 2 | 15:50:00 | 2 |
| 00:40:00 | 2 | 19:00:00 | 2 |
| 00:50:00 | 2 | 24:40:00 | 2 |

| **NRTRDE** | | **TAP3** | |
|---|---|---|---|
| **Часові проміжки** | **Кількість cdr_load_delay_ceiling** | **Часові проміжки** | **Кількість cdr_load_delay_ceilin** |
| 01:10:00 | 2 | 24:50:00 | 2 |
| 01:40:00 | 2 | 30:40:00 | 2 |
| 02:10:00 | 2 | 32:50:00 | 2 |
| 02:20:00 | 2 | 33:10:00 | 2 |
| 02:30:00 | 2 | 35:30:00 | 2 |
| 03:10:00 | 2 | 37:20:00 | 2 |
| 03:20:00 | 2 | | |
| 04:00:00 | 12 | | |
| **Загальна кількість** | 38 | **Загальна кількість** | 22 |
| **Середньозважене значення $T_{coll}$** | 02:15:47 | **Середньозважене значення $T_{coll}$** | 24:28:11 |

У процентному відношенні, розподілення виглядає наступним чином – 31% оповіщень були створені на основі NRTRDE файлів, що були завантажені протягом однієї години, 36% оповіщень на основі TAP3 файлів завантажених протягом 24 годин (рис. 3.7). Таке співвідношення дуже характерне з урахуванням специфіки інтервалів передачі такого типу даних. З деякими операторами мобільного зв’язку є можливість зменшити цей період, але основна проблема в тому, що цей канал даних та процеси підготовки файлів у оператора Б не є контрольованим з нашого боку.

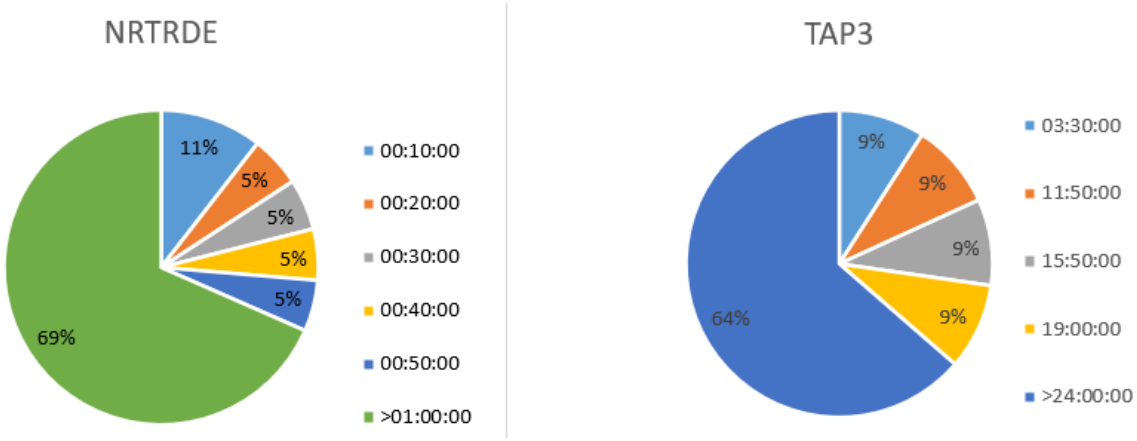



Рисунок 3.7 – кругова діаграма розподілення завантаження NRTRDE та TAP3 файлів.

Після включення у аналіз даних з Rhino CDR, співвідношення $T_{coll}$ та його середньозваженого значення для NRTRDE та TAP3 потерпають значних змін, які наведені у таблиці 3.10.

Таблиця 3.10 – кількість файлів та записів створених у день апробації при включені Rhino CDR у аналіз.

| **NRTRDE+Rhino CDR** | | **TAP3+Rhino CDR** | |
|---|---|---|---|
| **Часові проміжки** | **Кількість cdr_load_delay_ceiling** | **Часові проміжки** | **Кількість cdr_load_delay_ceilin** |
| 00:10:00 | 32 | 00:10:00 | 28 |
| 00:20:00 | 146 | 00:20:00 | 144 |
| 00:30:00 | 149 | 00:30:00 | 147 |
| 00:40:00 | 70 | 00:40:00 | 68 |
| 00:50:00 | 2 | 03:30:00 | 2 |
| 01:10:00 | 2 | 11:50:00 | 2 |
| 01:40:00 | 2 | 15:50:00 | 2 |
| 02:10:00 | 2 | 19:00:00 | 2 |
| 02:20:00 | 2 | 24:40:00 | 2 |
| 02:30:00 | 2 | 24:50:00 | 2 |
| 03:10:00 | 2 | 30:40:00 | 2 |

| NRTRDE+Rhino CDR | | TAP3+Rhino CDR | |
|---|---|---|---|
| **Часові проміжки** | **Кількість cdr_load_delay_ceiling** | **Часові проміжки** | **Кількість cdr_load_delay_ceilin** |
| 03:20:00 | 2 | 32:50:00 | 2 |
| 04:00:00 | 12 | 33:10:00 | 2 |
| | | 35:30:00 | 2 |
| | | 37:20:00 | 2 |
| **Загальна кількість** | 425 | **Загальна кількість** | 409 |
| **Середньозважене значення $T_{coll}$** | 00:36:21 | **Середньозважене значення $T_{coll}$** | 01:44:08 |

Такі зміни мають декілька причин, одна з яких не своєчасне завантаження NRTRDE, TAP3 файлів до системи. Це завантаження виражене тим, що CDR не попадають у інтервали моніторингу правил для виявлення шахрайства. Якщо Збільшити інтервали моніторингу, кількість оповіщень збільшиться, але ефективність у цьому випадку не зросте, та середньозважене значення збільшиться у рази. Також можна помітити, що збільшилися інтервали 30 та 40 хвилин, хоча теоретична середня затримка повинна бути 15 хвилин. Це пов’язане з тим, що частина файлів не попадає у періоди відпрацювання bash кодів або періоди завантаження у БД. Одним з можливих рішень є написанням одного коду, який виконував функції усіх трьох етапів одноразово з періодом у 5 хвилин.

У процентному відношенні, розподілення виглядає наступним чином – 94% оповіщень були створені на основі NRTRDE та Rhino CDR файлів, що були завантажені протягом однієї години, 95% оповіщень на основі TAP3 та Rhino CDR файлів завантажених протягом 1 години (рис. 3.8).

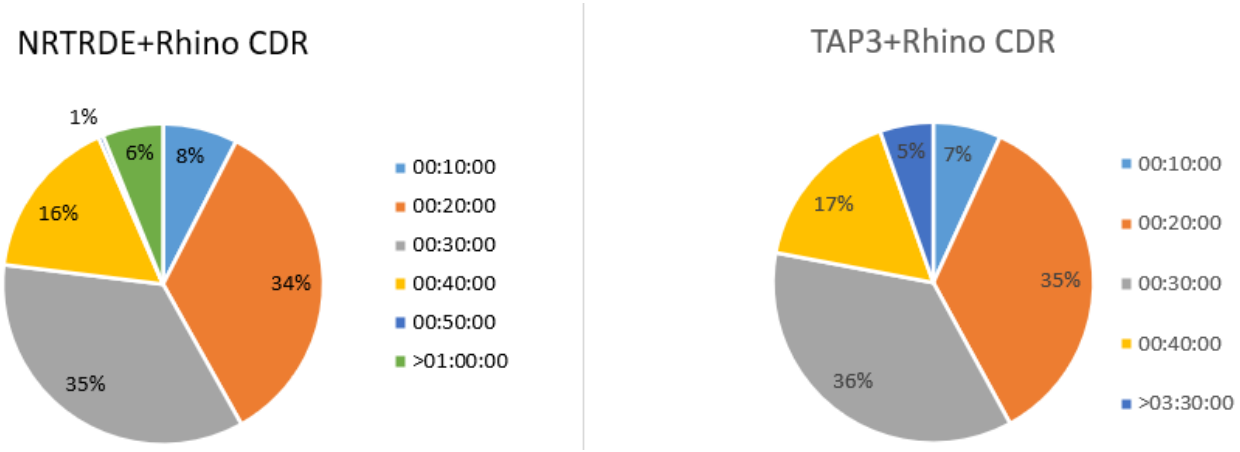



Рисунок 3.8 – кругова діаграма розподілення завантаження NRTRDE+Rhino CDR та TAP3+Rhino CDR файлів.

При порівнянні середньозваженого значення для NRTRDE+Rhino CDR та TAP3+Rhino CDR, було виявлено, що для NRTRDE затримка була зменшена у 3.7 раз, а для TAP3 у 14 разів.

Таблиця 3.11 – різниця середньзваженого $T_{coll}$ NRTRDE, NRTRDE+Rhino.

| | NRTRDE | NRTRDE+Rhino | Різниця |
|---|---|---|---|
| Середньозважене $T_{coll}$, hh24:mi:ss | 02:15:47 | 00:36:21 | 3,7 |

Таблиця 3.12 різниця середньзваженого $T_{coll}$ TAP3, TAP3+Rhino.

| | TAP3 | TAP3+Rhino | Різниця |
|---|---|---|---|
| Середньозважене $T_{coll}$, hh24:mi:ss | 24:28:11 | 01:44:08 | 14 |

## 3.4 Висновки до розділу

У першій частині даного розділу була наведена створена загальна схема обробки системи виявлення шахрайства та часові коефіцієнти ефективності кожного з етапів обробки трафіку. За допомогою середньозваженого значення

кожного коефіцієнту, є можливість отримати інформацію про найбільш впливові інтервали для кожного етапу. Даний підхід дозволяє проаналізувати роботу FMS у години навантаження або за останні 24 години.

У другій частині даного розділу, було наведено опис основних складових тестового середовища та технології, що використовувалися для реалізації, а саме базові функції UNIX як bash, NFS, віртуалізація мережевих функцій, тестова мережа четвертого покоління. Було проаналізовано складові AVP CDR Rhino, файлу з системи розрахунку надання послуг та створено таблицю з взаємозв'язком цих даних та доповнено розрахунком за надані послуги.

Наведено схему інтеграції різних джерел даних з системою виявлення шахрайства, а саме БД Oracle. Описаний процес, на базі bash кодів з використання Rhino SDK та Oracle ODI, насичення, трансформації та завантаження дозволяє сформулювати підхід для інших типів IMS послуг.

Розроблений алгоритм послідовної взаємодії IMS комутатора з системою виявлення шахрайства та розрахунку послуг, дозволив створити новий інтерфейс з завантаженням даних безпосередньо у базу даних FMS.

Результати апробації розробленого інтерфейсу свідчать про високу ефективність, що зменшила середньозважений час завантаження та легкість у інтеграції в порівнянні з безпосереднім збором за допомогою мережевих зондів. Апробація показала, що розроблені коефіцієнти ефективності можливо використовувати на будь-якому середовищі FMS у випадку оновлення джерела даних або зміни природи даних під час процесу моніторингу.

Отже, вперше розроблено метод оцінки ефективності системи розпізнання шахрайства, що ґрунтується на статичному методі з використанням вагомого коефіцієнту, на основі комплексного використання деталізованих записів, який дозволив зменшити середньовагоме значення часу затримки даних у 3.7 разів для NRTRDE та у 14 разів для TAP3.

# ВИСНОВКИ

У ході дисертаційного дослідження було проведено аналіз складових та технологій систем виявлення шахрайства, що дозволив визначити основні джерела інформації та підходи у обробці цих даних. На основі результату аналізу була створена загальна схема обробки та методика оцінки кожного етапу для визначення ефективності такої системи незалежно від типу технологій, на якій система побудована.

Було досліджено вплив шахрайства на інфокомунікаційну мережу та на оператора зв’язку. Визначено, що методи та технології реалізації того чи іншого виду шахрайства можуть між собою перетинатися та комбінуватися, що дозволяє їх визначити по декільком різним правилам виявлення. Зазначено, що методи шахрайства не обмежуються лише використанням властивостей обладнання мережі, а й людськими факторами такими як конфіденційною необачністю або використання співробітників для безпосереднього доступу до мережі.

Були вивчені основні види даних та методів збору інформації з мережі, серед яких існує два види файлів (NRTRDE, TAP3) обміну роумінговою інформацією між операторами зв’язку та метод безпосереднього моніторингу пакетів з мережі за допомогою мережевого зонду. Дійшли, висновку, що NRTRDE та TAP3, хоч і є стандартизованим форматом для обміну даними, має затримку що може варіюватися до 8 годин або навіть 40 днів. Розглянуті методі безпосереднього збору з мережі є дуже ефективними та надають надлишкову інформацію, що дозволяє їх використання у інших напрямках дослідження. Завдяки цьому, Отримав подальшого розвитку метод моніторингу віртуалізованого середовища з резервуванням, який на відміну від існуючих дозволив виявити дублікацію даних, встановлення додаткового мережного зонду під час розширення мережі для удосконалення моделі підтримки її інфраструктури.

Було наведено схему тестової мережі наближеної до традиційної на основі віртуалізованого середовища, що надає голосові сервіси за допомогою IMS. Розроблено схему взаємодії між OCS та IMS мережі A для насичення CDR з

наступним завантаженням у RDBMS Oracle, що належить FMS. Розроблено алгоритм взаємодії IMS комутатора з системою виявлення шахрайства та розрахунку послуг, наукова новизна якого полягає у використанні доступного bash кодування для форматування деталізованих записів, що базуються на застосуванні інструментів інтеграції даних, який дозволяє створити інтерфейс з наступним завантаженням інформації безпосередньо у базу даних системи моніторингу.

Вперше розроблено метод оцінки ефективності системи розпізнання шахрайства, що ґрунтується на статичному методі з використанням вагомого коефіцієнту, на основі комплексного використання деталізованих записів, який дозволив зменшити середньовагоме значення часу затримки даних у 3.7 разів для NRTRDE та у 14 разів для TAP3.

Практичне застосування розробленої методики: реалізовано алгоритм взаємодії на базі bash кодів та Oracle ODI. Проведено експериментальне застосування розробленого інтерфейсу у комбінації з традиційними форматами NRTRDE, TAP3 з правилами визначення великої кількості коротких або великої тривалості дзвінків в залежності від напрямку отримувача.

Наукові проблеми та напрями подальших досліджень включають:

1. Оптимізація реалізованого ETL для затримки у кожному етапі підготовки CDR для зменшення середньозваженого коефіцієнту завантаження даних ($T_{coll}$).

2. Розширення типу IMS сервісів для спостереження USSD та SMS;

3. Розробка інтерфейсу для відстеження сервісів з мереж канальної комутації каналів у випадку сценаріїв взаємодії IMS абонента з абонентом 2G/3G або між абонентами 2G/3G;

4. Інтеграція з мережами пакетної комутації для відстеження ігрових, стрімінгових або інших типів сервісів;

5. Розширення застосування шляхом інтеграції з корпоративним фаєрволом для блокування сесій та іншими системи контролю.

Результати цієї дисертаційної роботи відкривають широкі перспективи для подальших досліджень у галузі безпеки інформаційних мереж, сприяють у розвитку нових методів захисту мережі та існуючих методів захисту до їх оптимізації.

Результати наукового дослідження були використані та впроваджені у проекті на підприємстві ТОВ "ІНФОПУЛЬС УКРАЇНА".

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алтинніков Д. Є., Шевченко О. О., Бердник І. І., Зуб О. В., Сагайдак В. А., «Використання Java--анотацій як інструменту надання API», *Зв'язок*, № 4(152), с. 56–59, 2021. URL: https://doi.org/10.31673/2412-9070.2021.045659

2. Сагайдак В. А., Сеньков О. В., «Huawei Genex Discovery – інструмент виявлення великих даних для аналізу безпроводової мережі», *Зв'язок*, № 4(158), с. 34–41, 2022. URL: https://doi.org/10.31673/2412-9070.2022.043441

3. Сагайдак В. А., Лисенко М. М., Сеньков О. В., «Шахрайство у сфері телекомунікацій та його вплив на бізнес операторів зв'язку», *Зв'язок*, № 6(160), с. 17–20, 2022. URL: https://doi.org/10.31673/2412-9070.2022.061720

4. Сагайдак В. А., «Огляд систем розпізнання шахрайства та розробка коефіцієнтів для визначення їх ефективності», *OTH*, № X (), с. XX-XX , 202X.

5. Сагайдак В. А., «Огляд методів збору даних на мережі телекомунікацій за допомогою мережевого зонду», *Телекомунікаційні та інформаційні технології*, № 2 (), с. XX-XX , 2024.

6. IX Науково-технічна конференція студентів та молодих вчених факультету Інформаційних технологій «Сучасні інфокомунікаційні технології»; Система для аналізу та моніторингу радіопокриття базових станцій; 11 грудня 2020, Державний університету телекомунікацій, м. Київ.

7. IV Всеукраїнська науково-практична конференція «Telecommunication: problems and innovation»; SMS fraud realization and recognition methods; 30 травня 2023 р., Державний університету телекомунікацій, м. Київ.

8. V Всеукраїнська науково-практична конференція «Telecommunication: problems and innovation»; TAP3 and NRTRDE CDR transfer formats; 20 грудня 2023р., Державний університет інформаційно-комунікаційних технологій, м. Київ.

9. XIII міжнародна науково технічна конференція The 13th International Scientific Conference «ITSEC»; Rhino IMS CDR APV fields for network and subscriber identification; 9-11 травня 2024 р., Львівський національний університет імені Івана Франка, м. Львів.

10. scribd [електронний ресурс]// SC1002 HUAWEI SmartCare SEQ Analyst & NetProbe Technical Slides V2.4 – Режим доступу - https://www.scribd.com/doc/273930937/SC1002-HUAWEI-SmartCare-SEQ-Analyst-NetProbe-Technical-Slides-V2-4

11. Carrier Huawei [електронний ресурс]// best Network - Режим доступу - https://carrier.huawei.com/en/products/service-and-software/best-network

12. Support Huawei [електронний ресурс]// GENEX Discovery Product Documentation(CPI)(PS) – Режим доступу - https://support.huawei.com/hedex/hdx.do?docid=DOC1000269102&path=PBI1-7275736/PBI1-9855706/PBI1-7275887/PBI1-21051528

13. scribd [електронний ресурс]// Genex Discovery Tutorial – Режим доступу - https://scribd.com/presentation/440716343/Genex-Discovery-Tutorial

14. Head First Java, 2nd Edition by Kathy Sierra, Bert Bates. 2005. С. 233–249.

15. Java: The Complete Reference, Eleventh Edition 11th Edition by Herbert Schildt. 2018. С. 301–323.

16. Java Professional Library by David Flanagan. 2000. С. 171–188.

17. Java Developer’s Reference by Bryan Morgan, Michael Morrison, Michael T. Nygard, Dan Joshi, Tom Trinko, Mike Cohn. 1996. С. 201–212.

18. Effective Java 3rd Edition by Joshua Bloch. 2017. С. 99–111.

19. Thinking in Java 4th Edition by Bruce Eckel. 2006. С. 247–255.

20. Gigamon [електронний ресурс]// Understanding international telecoms fraud - Режим доступу - https://www.gigamon.com/content/dam/resource-library/english/technology-partner-solution-brief/js-argyle-data-gigamon-fraud-detection-big-data-analytics.pdf

21. Vanilla+ [електронний ресурс]// Argyle Data and Gigamon to deliver real-time fraud detection and analytics for communications service providers - Режим доступу - https://www.vanillaplus.com/2015/03/04/5865-argyle-data-and-gigamon-to-deliver-real-time-fraud-detection-and-analytics-for-communications-service-providers/

22. Gigamon [електронний ресурс]// Understanding Network TAPs – The First Step to Visibility – Режим доступу - https://www.gigamon.com/resources/resource-library/white-paper/understanding-network-taps-first-step-to-visibility.html

23. Huawei [електронний ресурс]// Example for Configuring Local Port Mirroring (1:1 Mirroring) - S600-E Series Switches Typical Configuration Examples – Режим доступу - https://support.huawei.com/enterprise/en/doc/EDOC1000141870/4b4d7f54/example-for-configuring-local-port-mirroring-11-mirroring

24. Huawei [електронний ресурс]// Configuring Local Port Mirroring - CloudEngine S8700 V600R022C00 Configuration Guide - System Monitoring - Режим доступу - https://support.huawei.com/enterprise/en/doc/EDOC1100277360/86abdfc8

25. BICS [електронний ресурс]// Understanding international telecoms fraud - Режим доступу - https://www.bics.com/wp-content/uploads/2022/02/Telco-Fraud-Whitepaper.pdf

26. CFCA [електронний ресурс]// Putting telecom fraud loss into perspective… - Режим доступу - https://cfca.org/putting-telecom-fraud-loss-into-perspective/

27. EUROPOL [електронний ресурс]// Telecommunications Fraud – Режим доступу - https://www.europol.europa.eu/operations-services-and-innovation/public-awareness-and-prevention-guides/telecommunications-fraud

28. SEON [електронний ресурс]// 11 Types of Telecommunications Fraud: How to Detect & Prevent It – Режим доступу - https://seon.io/resources/telecommunications-fraud-detection-and-prevention/

29. Mobileum [електронний ресурс]// Telecom Wholesale Fraud – Режим доступу - https://www.mobileum.com/products/risk-management/fraud-management/telecom-wholesale-fraud/

30. Arkose Labs [електронний ресурс]// International Revenue Share Fraud (IRSF): What it is and How to Stop it – Режим доступу - https://www.arkoselabs.com/explained/international-revenue-share-fraud/

31. Mobileum [електронний ресурс]// Telecom Wholesale Fraud – Режим доступу - https://www.mobileum.com/products/risk-management/fraud-management/bypass-fraud/

32. Mobileum [електронний ресурс]// Telecom Wholesale Fraud – Режим доступу - https://www.mobileum.com/products/risk-management/fraud-management/revenue-share-fraud/

33. EUROPOL [online]// Takedown of SMS-based FluBot spyware infecting Android phones – Available - https://www.europol.europa.eu/media-press/newsroom/news/takedown-of-sms-based-flubot-spyware-infecting-android-phones

34. SEON [online]// 11 Types of Telecommunications Fraud: How to Detect & Prevent It – Available - https://seon.io/resources/telecommunications-fraud-detection-and-prevention/

35. Nick vs Networking [online]// An intro to GSMA TAP3 Files – Available - https://nickvsnetworking.com/an-intro-to-gsma-tap3-files/

36. GSMA [online]// Use of TAP for the Single IMSI Wholesale Billing Interface – Available - https://www.gsma.com/get-involved/working-groups/interoperability-data-specifications-and-settlement-group/standardised-b2b-interfaces-specified-by-ids/open-standards-specifications/tap3-open-standard-download-form

37. The MACH Blog [online]// Introducing TAP-NRTRDE Reconciliation– Available - https://machinsights.wordpress.com/2013/04/09/introducing-tap-nrtrde-reconciliation/

38. GSMA [online]// Use of NRTRDE for the Single IMSI Fraud Interface – Available - https://www.gsma.com/get-involved/working-groups/interoperability-data-specifications-and-settlement-group/standardised-b2b-interfaces-specified-by-ids/open-standards-specifications/tap3-open-standard-download-form

39. Christopher Cox. An introduction to LTE LTE, LTE-advanced, SAE, VoLTE and 4G mobile communications. Publisher: Wiley, 2014. 449 p.

40. Sentinel VoLTE Architecture, URL: https://docs.rhino.metaswitch.com/ocdoc/books/sentinel-volte-documentation/4.1/sentinel-volte-architecture/index.html (application date: 16.04.2024).

41. Sentinel VoLTE Administration Guide. URL: https://docs.rhino.metaswitch.com/ocdoc/books/sentinel-volte-documentation/4.1/sentinel-volte-administration-guide/index.html (application date: 16.04.2024).

42. ETSI TS 132 299. URL: https://www.etsi.org/deliver/etsi_ts/132200_132299/132299/12.11.00_60/ts_132299v121100p.pdf (application date: 16.04.2024).

43. Real-Time Fraud Detection and Analytics using Hadoop and Machine Learning. (2015, February 15). Network-Level Intelligence for Observability Tools | Gigamon. URL: http://surl.li/tkbjz

44. Real-Time Fraud Analytics Hadoop Application. (2014, November). Cloudera | The hybrid data company. URL: http://surl.li/tkbmp

45. Intelligent Fraud Monitoring | AWS Solutions for Telecommunications | AWS Solutions Library. (n.d.). Amazon Web Services, Inc. URL: http://surl.li/tkbmu

46. Telecom Fraud Management | Telecom Fraud Detection | Telco Risk. (n.d.). Subex. URL: http://surl.li/tkbmz

47. PPT - Fraud Management and Operations Training PowerPoint Presentation - ID:1050298. (n.d.). SlideServe. URL: http://surl.li/tkbne

48. CVidya Launches FraudView® Version 9. (2010, October 13). Newswire | Press Release Distribution | Media Outreach Platform. URL: http://surl.li/tkbnm

49. Amazon S3 - Cloud Object Storage - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/s3/?did=ap_card&trk=ap_card.

50. Amazon SageMaker Feature Store for machine learning (ML) – Amazon Web Services. Amazon Web Services, Inc. URL: https://aws.amazon.com/sagemaker/feature-store/.

51. Apache Hadoop Architecture Explained (In-Depth Overview). Knowledge Base by phoenixNAP. URL: https://phoenixnap.com/kb/apache-hadoop-architecture-explained.

52. Apache Hadoop on Amazon EMR - Big Data Platform - Amazon Web Services. Amazon Web Services, Inc. URL: https://aws.amazon.com/emr/features/hadoop/.

53. API Management - Amazon API Gateway - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/api-gateway/?did=ap_card&trk=ap_card.

54. Data Stream Processing - Amazon Kinesis - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/kinesis/?did=ap_card&trk=ap_card.

55. Data Wrangling Tool - Amazon SageMaker Data Wrangler - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/sagemaker/data-wrangler/?nc2=type_a.

56. ETL Service - Serverless Data Integration - AWS Glue - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/glue/.

57. Interactive SQL - Amazon Athena - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/athena/.

58. Machine Learning Service - Amazon SageMaker - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/sagemaker/?did=ap_card&trk=ap_card.

59. Manage your endpoints - Amazon SageMaker. Amazon Web Services, Inc. URL: https://docs.aws.amazon.com/sagemaker/latest/dg/realtime-endpoints-manage.html.

60. Monitor data and model quality - Amazon SageMaker. Amazon Web Services, Inc. URL: https://docs.aws.amazon.com/sagemaker/latest/dg/model-monitor.html.

61. Register and Deploy Models with Model Registry - Amazon SageMaker. Amazon Web Services, Inc. URL: https://docs.aws.amazon.com/sagemaker/latest/dg/model-registry.html.

62. Serverless Computing - AWS Lambda Features - Amazon Web Services. Amazon Web Services, Inc. URL: https://aws.amazon.com/lambda/features/?pg=ln&sec=hs.

63. Serverless Function, FaaS Serverless - AWS Lambda - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/lambda/?did=ap_card&trk=ap_card.

64. Shukla S., Kukade V., Mujawar S. Big Data: Concept, Handling and Challenges: An Overview. International Journal of Computer Applications. 2015. Vol. 114, no. 11. P. 6–9. URL: https://doi.org/10.5120/20020-1537.

65. SoK: Fraud in Telephony Networks / M. Sahin et al. 2017 IEEE European Symposium on Security and Privacy (EuroS&P), Paris, 26–28 April 2017. 2017. URL: https://doi.org/10.1109/eurosp.2017.40.

66. Telecommunications fraud increased 12% in 2023 equating to an estimated $38.95 billion lost to fraud. CFCA. URL: https://cfca.org/telecommunications-fraud-increased-12-in-2023-equating-to-an-estimated-38-95-billion-lost-to-fraud/.

67. What Is Amazon Kinesis Data Streams? - Amazon Kinesis Data Streams. Amazon Web Services, Inc. URL: https://docs.aws.amazon.com/streams/latest/dev/introduction.html.

68. What is NoSQL? | Nonrelational Databases, Flexible Schema Data Models | AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/nosql/?nc1=h_ls.

69. Connectivity and Knowledge Modules Guide for Oracle Data Integrator. Oracle Help Center. URL: https://docs.oracle.com/en/middleware/fusion-middleware/data-integrator/12.2.1.4/odikm/index.html.

70. Developing Integration Projects with Oracle Data Integrator. Oracle Help Center. URL: https://docs.oracle.com/en/middleware/fusion-middleware/data-integrator/12.2.1.4/odidg/creating-integration-project.html#GUID-A016BDD9-6EEE-4B2B-A1B8-027054574F02.

71. Sentinel Express 4.1 :: Sentinel Overview and Concepts :: New Session Counters. Metaswitch Documentation. URL: https://docs.rhino.metaswitch.com/ocdoc/books/sentinel-documentation/4.1/sentinel-

overview-and-concepts/sentinel-sip-enhancements/inside-sentinel-sip/session-counters-sip.html#session-counter-structure.

72. Sentinel VoLTE 4.0.0 :: Sentinel VoLTE Administration Guide :: Custom Headers. Metaswitch Documentation. URL: https://docs.rhino.metaswitch.com/ocdoc/books/sentinel-volte-documentation/4.0.0/sentinel-volte-administration-guide/session-processing/custom-headers/index.html.

73. Sentinel VoLTE 4.1 :: Sentinel VoLTE Administration Guide :: AVP CDR Format. Metaswitch Documentation. URL: https://docs.rhino.metaswitch.com/ocdoc/books/sentinel-volte-documentation/4.1/sentinel-volte-administration-guide/charging-information/cdr-formats/avp-cdr-format.html.

74. Sentinel VoLTE 4.1 :: Sentinel VoLTE Administration Guide :: Charging Content AVPs. Metaswitch Documentation. URL: https://docs.rhino.metaswitch.com/ocdoc/books/sentinel-volte-documentation/4.1/sentinel-volte-administration-guide/charging-information/charging-content-avps/index.html#populated-avps-in-the-mmtel-information-avp.

75. Sentinel VoLTE 4.1 :: Sentinel VoLTE Administration Guide :: Ro Interface AVPs. Metaswitch Documentation. URL: https://docs.rhino.metaswitch.com/ocdoc/books/sentinel-volte-documentation/4.1/sentinel-volte-administration-guide/charging-information/ro-interface-avps/index.html.

76. Sentinel VoLTE 4.1 :: Sentinel VoLTE Administration Guide :: Running List CDRs. Metaswitch Documentation. URL: https://docs.rhino.metaswitch.com/ocdoc/books/sentinel-volte-documentation/4.1/sentinel-volte-administration-guide/charging-information/working-with-cdrs/running-list-cdrs.html.

77. Sentinel VoLTE 4.1 :: Sentinel VoLTE Administration Guide :: Sentinel AVP definitions. Metaswitch Documentation. URL: https://docs.rhino.metaswitch.com/ocdoc/books/sentinel-volte-

documentation/4.1/sentinel-volte-administration-guide/charging-information/charging-content-avps/sentinel-avp-definitions.html.

78. Sentinel VoLTE 4.1 :: Sentinel VoLTE Administration Guide :: Working with CDRs. Metaswitch Documentation. URL: https://docs.rhino.metaswitch.com/ocdoc/books/sentinel-volte-documentation/4.1/sentinel-volte-administration-guide/charging-information/working-with-cdrs/index.html.

79. Understanding Oracle Data Integrator. Oracle Help Center. URL: https://docs.oracle.com/en/middleware/fusion-middleware/data-integrator/12.2.1.4/odiun/understanding-oracle-data-integrator-concepts.html#GUID-68CB2BEC-0448-409E-85FE-F176BB72983F.

80. Virtualization Software Requirements. Cisco: Software, Network, and Cybersecurity Solutions - Cisco. URL: https://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/uc_system/virtualization/virtualization-software-requirements.html.

81. A Survey of BGP Security Issues and Solutions / K. Butler et al. Proceedings of the IEEE. 2010. Vol. 98, no. 1. P. 100–122. URL: https://doi.org/10.1109/jproc.2009.2034031

82. Becker R. A., Volinsky C., Wilks A. R. Fraud Detection in Telecommunications: History and Lessons Learned. Technometrics. 2010. Vol. 52, no. 1. P. 20–33. URL: https://doi.org/10.1198/tech.2009.08136

83. Breaking and Fixing VoLTE / H. Kim et al. CCS'15: The 22nd ACM Conference on Computer and Communications Security, Denver Colorado USA. New York, NY, USA, 2015. URL: https://doi.org/10.1145/2810103.2813718

84. Clayton R. Can CLI be trusted?. Information Security Technical Report. 2007. Vol. 12, no. 2. P. 74–79. URL: https://doi.org/10.1016/j.istr.2007.04.002

85. Dabrowski A., Petzl G., Weippl E. R. The Messenger Shoots Back: Network Operator Based IMSI Catcher Detection. Research in Attacks, Intrusions, and Defenses. Cham, 2016. P. 279–302. URL: https://doi.org/10.1007/978-3-319-45719-2_13

86. Dialing Back Abuse on Phone Verified Accounts / K. Thomas et al. CCS'14: 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale Arizona USA. New York, NY, USA, 2014. URL: https://doi.org/10.1145/2660267.2660321

87. FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild / Z. Li et al. Network and Distributed System Security Symposium, San Diego, CA. Reston, VA, 2017. URL: https://doi.org/10.14722/ndss.2017.23098

88. Goldstein E. Best of 2600: A Hacker Odyssey. Wiley & Sons, Incorporated, John, 2008.

89. Henecka W., Roughan M. Privacy-Preserving Fraud Detection Across Multiple Phone Record Databases. IEEE Transactions on Dependable and Secure Computing. 2015. Vol. 12, no. 6. P. 640–651. URL: https://doi.org/10.1109/tdsc.2014.2382573

90. IMSI-catch me if you can / A. Dabrowski et al. the 30th Annual Computer Security Applications Conference, New Orleans, Louisiana, 8–12 December 2014. New York, New York, USA, 2014. URL: https://doi.org/10.1145/2664243.2664272

91. Inside the scam jungle: a closer look at 419 scam email operations / J. Isacenkova et al. EURASIP Journal on Information Security. 2014. Vol. 2014, no. 1. URL: https://doi.org/10.1186/1687-417x-2014-4.

92. Loughney J., Pastor-Balbas J. Security Considerations for Signaling Transport (SIGTRAN) Protocols / ed. by M. Tuexen. RFC Editor, 2004. URL: https://doi.org/10.17487/rfc3788.

93. Phoneypot: Data-driven Understanding of Telephony Threats / P. Gupta et al. Network and Distributed System Security Symposium, San Diego, CA. Reston, VA, 2015. URL: https://doi.org/10.14722/ndss.2015.23176

94. Phonion: Practical Protection of Metadata in Telephony Networks / S. Heuser et al. Proceedings on Privacy Enhancing Technologies. 2017. Vol. 2017, no. 1. P. 170–187. URL: https://doi.org/10.1515/popets-2017-0011.

95. Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems / A. Shaik et al. Network and Distributed System Security

Symposium, San Diego, CA. Reston, VA, 2016. URL: https://doi.org/10.14722/ndss.2016.23236.

96. Sahin M., Francillon A. Over-The-Top Bypass. CCS'16: 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna Austria. New York, NY, USA, 2016. URL: https://doi.org/10.1145/2976749.2978334.

97. Signaling Vulnerabilities in Wiretapping Systems / M. Sherr et al. IEEE Security and Privacy Magazine. 2005. Vol. 3, no. 6. P. 13–25. URL: https://doi.org/10.1109/msp.2005.160.

98. SoK: Everyone Hates Robocalls: A Survey of Techniques Against Telephone Spam / H. Tu et al. 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, 22–26 May 2016. 2016. URL: https://doi.org/10.1109/sp.2016.27.

99. The role of phone numbers in understanding cyber-crime schemes / A. Costin et al. 2013 Eleventh Annual Conference on Privacy, Security and Trust (PST), Tarragona, Spain, 10–12 July 2013. 2013. URL: https://doi.org/10.1109/pst.2013.6596056.

100. You Can Call but You Can't Hide: Detecting Caller ID Spoofing Attacks / H. Mustafa et al. 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Atlanta, GA, USA, 23–26 June 2014. 2014. URL: https://doi.org/10.1109/dsn.2014.102

101. Authenticated Identity Management in the Session Initiation Protocol (SIP) / J. Peterson et al. RFC Editor, 2018. URL: https://doi.org/10.17487/rfc8224

102. Peterson J., Turner S. Secure Telephone Identity Credentials: Certificates. RFC Editor, 2018. URL: https://doi.org/10.17487/rfc8226

103. System Administrator's Guide. Oracle Help Center. URL: https://docs.oracle.com/en/industries/communications/billing-revenue/15.0/sys-admin/synchronization-utilities1.html#GUID-869336C8-ECF6-4B12-A6AF-99C6A7D0261E.

104. Definition of Tier 1 network. PCMag. URL: https://www.pcmag.com/encyclopedia/term/tier-1-network.

105. Tier 0: A new category of telecom operators is born. The reign of Tier 1 operators such as Orange, Vodafone, Telefonica, and Deutsche Telekom is over - Strand Consult. Strand Consult. URL: https://strandconsult.dk/tier-0-a-new-category-of-telecom-operators-is-born-the-reign-of-tier-1-operators-such-as-orange-vodafone-telefonica-and-deutsche-telekom-is-over/.

106. Olsson M., Mulligan C. EPC and 4G Packet Networks: Driving the Mobile Broadband Revolution. Academic Press, 2018. 624 p.

107. 1. D. Patel C., J. Shah A. Cost Model for Planning, Development and Operation of a Data Center. ResearchGate. URL: https://www.researchgate.net/publication/245808024_Cost_Model_for_Planning_Development_and_Operation_of_a_Data_Center (date of access: 11.06.2024).

108. Сагайдак В. А., Сачук О. В., «Розроблення методики транскрибації на основі нейронних мереж», *Зв'язок*, № 2(168), с. 23–26, 2024. URL: https://doi.org/10.31673/2412-9070.2024.022326

# ДОДАТОК А

## CDR конфігурація sentinel-volte-gsm-config.yaml

```
# Configuration for CDRs
cdr:
  # If present, interim CDRs are enabled. If Diameter Rf has been enabled, this is required.
  interim-cdrs:
    # Enable CDRs to go to the local filesystem
    # Diameter Rf is selected separately
    write-cdrs-in-filesystem: false

    # Indicates whether or not to write CDRs on SDP changes.
    write-cdr-on-sdp-change: false

    # The maximum duration (in seconds) between timer driven interim CDRs.
    # Setting this to zero will disable timer based interim CDRs.
    interim-cdrs-period-seconds: 300

  # Enable session CDRs.
  session-cdrs-enabled: true
```

## bash код binary_to_human_read.sh

```
ls /apps/Rhino_TAS/cdr/cdr_* | xargs -n 1 basename >/home/testuser/files_for_read.txt
INFILE=/home/testuser/files_for_read.txt
while read -r LINE
do
  /home/testuser/sentinel-volte-sdk/tools/list-cdrs/list-cdrs.sh --output-file /fraud_detection/Rhino/raw/"$LINE".txt /apps/Rhino_TAS/cdr/"$LINE"
  mv /apps/Rhino_TAS/cdr/"$LINE" /apps/Rhino_TAS/cdr/done/
done < "$INFILE"

chmod 755 /fraud_detection/Rhino/raw/*
```

## cron bash коду binary_to_human_read.sh

```
#human readable crontab
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
```

```
# * * * * * user-name command to be executed
#Receiving
### Receiving All
*/5 * * * * testuser /home/testuser/binary_to_human_read.sh
```

**bash код export_product.sh**

```
/BRM_home/bin/pin_export_price -product
cp /BRM_home/bin/export_product.csv /fraud_detection/Price/in
chmod 755 /fraud_detection/Price/in/export_product.csv
```

**cron bash коду export_product.sh**

```
#Price_export crontab
# .---------------- minute (0 - 59)
# | .------------- hour (0 - 23)
# | | .---------- day of month (1 - 31)
# | | | .------- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
#Receiving
### Receiving All
* 1 * * * export_user /home/export_user/export_price.sh
```

**bash код convert.sh**

```
#!/bin/bash
ls /media/sf_fraud_detection/Rhino/raw/cdr_*.txt | xargs -n 1 basename
>/home/oracle/files_for_convert.txt
INFILE=/home/oracle/files_for_convert.txt
while read -r LINE
do
        grep hostname /media/sf_fraud_detection/Rhino/raw/"$LINE"
>>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed -i s/hostname=/"/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        echo "$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        grep OC-Service-Type /media/sf_fraud_detection/Rhino/raw/"$LINE"
>>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed -i s/'OC-Service-Type(Ext,Ext)'/"/
/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
```

```
grep OC-Call-Type /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'OC-Call-Type(Ext,Ext)'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep User-Session-Id /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'User-Session-Id'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep Originating-IOI /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'Originating-IOI'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep Terminating-IOI /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'Terminating-IOI'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep OC-Start-Time /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'OC-Start-Time'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep OC-Session-Start-Time /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'OC-Session-Start-Time'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep OC-Session-Established-Time /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'OC-Session-Established-Time'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep OC-End-Time /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'OC-End-Time'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep OC-Session-End-Time /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'OC-Session-End-Time'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo UTC+0 >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
```

```
grep Subscription-Id-Data /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'Subscription-Id-Data'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep Calling-Party-Address /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'Calling-Party-Address'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep Called-Party-Address /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'Called-Party-Address'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep Called-Asserted-Identity /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'Called-Asserted-Identity'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep Requested-Party-Address /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'Requested-Party-Address'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep 'OC-Call-Id(Ext,Ext)' /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'OC-Call-Id(Ext,Ext)'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep Role-Of-Node /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'Role-Of-Node'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep Node-Functionality /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'Node-Functionality'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
grep OC-Charging-Instance-Name /media/sf_fraud_detection/Rhino/raw/"$LINE" >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
sed -i s/'OC-Charging-Instance-Name'/''/ /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
```

```
        echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        grep 'OC-OCS-Session-Termination-Cause(Ext,Ext)'
/media/sf_fraud_detection/Rhino/raw/"$LINE"
>>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed -i s/'OC-OCS-Session-Termination-Cause(Ext,Ext)'/''/
/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        grep IMS-Communication-Service-Identifier
/media/sf_fraud_detection/Rhino/raw/"$LINE"
>>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed -i s/'IMS-Communication-Service-Identifier'/''/
/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        grep IMS-Charging-Identifier /media/sf_fraud_detection/Rhino/raw/"$LINE"
>>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed -i s/'IMS-Charging-Identifier'/''/
/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        echo , >>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        grep Equipment-Info-Value /media/sf_fraud_detection/Rhino/raw/"$LINE"
>>/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed -i s/'Equipment-Info-Value'/''/
/media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed s/' '//g -i /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed 's/^[ \t]*//g' -i /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed s/'],'//g -i /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed s/']'//g -i /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed 's/[][]//g' -i /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        sed 's/([^]]*)//g' -i /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
        cat /media/sf_fraud_detection/Rhino/raw/temp_"$LINE" | tr -d '\r\n' >
/media/sf_fraud_detection/Rhino/in/"$LINE".csv
        #mv /media/sf_fraud_detection/Rhino/raw/"$LINE"
/media/sf_fraud_detection/Rhino/temp/
        mv /media/sf_fraud_detection/Rhino/raw/temp_"$LINE"
/media/sf_fraud_detection/Rhino/temp/
done < "$INFILE"
```

**cron bash коду convert.sh**

```
#human readable crontab
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
```

```
# | | | | |
# * * * * * user-name command to be executed
#Receiving
### Receiving All
*/5    *      *      *      *    etluser /home/etluser/convert.sh
```

**xml конфігурація схеми ODI взаємодії файлів export_product.csv та cdr_101_*.txt.csv**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SunopsisExport>
<Admin RepositoryVersion="05.02.02.07" IsLegacyIdCompatible="false" />
<Encryption algorithm="AES" keyLength="128"
exportKeyHash="jWj4qtm6gDIXKaKm8m+BVGJbAp2YrfkpTTwfxSDWzbM="
keyVect="DYbYYSuPhLjoP2b1jgKRJw==" exportKeySalt="e0867f15-4b6e-48d3-
9d22-79a9f8554343" containsCipherText="false"/>
------------------------(Наповнення таблиці fraud.ur_ims_records) ----------------
        <Field name="DefTxt" type="java.lang.String"><![CDATA[
INSERT
 /*+  APPEND PARALLEL  */
 INTO <?= odiRef.getObjectName("L", "UR_IMS_RECORDS", "DB-Fraud", "D") ?>
 (
   CDR_ID ,
   LOADING_TIME ,
   SOURCE_NAME ,
   FILE_NAME ,
   SERVICE_NAME ,
   SERVICE_TYPE ,
   SESSION_ID ,
   ORIGINATED_NETWORK ,
   DESTINATION_NETWORK ,
   CURRENCY ,
   MIN_PRICE ,
   PRICE_SDR ,
   START_CALL_TIME ,
   SESSION_START_TIME ,
   SESSION_ESTABLISHMENT_TIME ,
   END_CALL_TIME ,
   SESSION_END_TIME ,
   SESSION_DURATION ,
   UTC_OFFSET ,
   SERVED_MSISDN ,
   ORIGINATED_NUMBER ,
   DESTINATION_NUMBER ,
```

```
    DESTINATION_IMEI ,
    ID_THIRD_PARTY_NUM ,
    ID_DIALED_NUMBER ,
    NT_CHARGE_ID ,
    SERVING_NODE_ROLE ,
    SERVING_NODE_FUNCTION ,
    CHARGING_INSTANCE ,
    NT_TERMINATION_CAUSE ,
    SESSION_ICSI ,
    SESSION_ICI
  )
SELECT
  cdr_id_seq.NEXTVAL ,
  SYSDATE ,
  CDR_A.SOURCE_NAME ,
  CDR_A.FILE_NAME ,
  CDR_A.SERVICE_NAME ,
  CDR_A.SERVICE_TYPE ,
  CDR_A.SESSION_ID ,
  CDR_A.ORIGINATED_NETWORK ,
  CDR_A.DESTINATION_NETWORK ,
  EXP_A.CURRENCY ,
  EXP_A.MIN_PRICE ,
(EXP_A.MIN_PRICE*(CDR_A.END_CALL_TIME-
CDR_A.START_CALL_TIME)/60000) ,
  (to_date('1970-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS') +
numtodsinterval((CDR_A.START_CALL_TIME/1000),'SECOND')) ,
  (to_date('1970-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS') +
numtodsinterval((CDR_A.SESSION_START_TIME/1000),'SECOND')) ,
  (to_date('1970-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS') +
numtodsinterval((CDR_A.SESSION_ESTABLISHMENT_TIME/1000),'SECOND')) ,
  (to_date('1970-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS') +
numtodsinterval((CDR_A.END_CALL_TIME/1000),'SECOND')) ,
  (to_date('1970-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS') +
numtodsinterval((CDR_A.SESSION_END_TIME/1000),'SECOND')) ,
  (CDR_A.END_CALL_TIME-CDR_A.START_CALL_TIME)/60000 ,
  CDR_A.UTC_OFFSET ,
  CDR_A.SERVED_MSISDN ,
  CDR_A.ORIGINATED_NUMBER ,
  CDR_A.DESTINATION_NUMBER ,
  CDR_A.EQUIPMENT_INFO_VALUE ,
  CDR_A.ID_THIRD_PARTY_NUM ,
  CDR_A.ID_DIALED_NUMBER ,
  CDR_A.NT_CHARGE_ID ,
```

```
 CDR_A.SERVING_NODE_ROLE ,
 CDR_A.SERVING_NODE_FUNCTION ,
 CDR_A.CHARGING_INSTANCE ,
 CDR_A.NT_TERMINATION_CAUSE ,
 CDR_A.SESSION_ICSI ,
 CDR_A.SESSION_ICI
FROM
 <?= odiRef.getObjectName("L", "%COL_PRF0EXP", "DB-Fraud", "W") ?> EXP_A
INNER JOIN <?= odiRef.getObjectName("L", "%COL_PRF1CDR", "DB-Fraud", "W")
?> CDR_A
  ON CDR_A.SERVICE_NAME=EXP_A.SERVICE_NAME and
EXP_A.SERVICE_TYPE=CDR_A.SERVICE_TYPE and
CDR_A.ORIGINATED_NETWORK=EXP_A.IOI
   ]]></Field>
      <Field name="ExeChannel" type="java.lang.String"><![CDATA[J]]></Field>
      <Field name="GlobalId" type="java.lang.String"><![CDATA[59df67c6-6851-
4075-9962-6578fb86967c]]></Field>
      <Field name="IndErr" type="java.lang.String"><![CDATA[0]]></Field>
      <Field name="IndLogFinalCmd"
type="java.lang.String"><![CDATA[0]]></Field>
      <Field name="IndLogMethod" type="java.lang.String">null</Field>
      <Field name="IndLogNb" type="java.lang.String"><![CDATA[I]]></Field>
      <Field name="LogLevDet" type="java.lang.String"><![CDATA[5]]></Field>
      <Field name="MapTaskType"
type="java.lang.String"><![CDATA[EM]]></Field>
      <Field name="Nno" type="com.sunopsis.sql.DbInt"><![CDATA[10]]></Field>
      <Field name="OrdTrt"
type="com.sunopsis.sql.DbInt"><![CDATA[10]]></Field>
      <Field name="ParScenTaskNo"
type="com.sunopsis.sql.DbInt"><![CDATA[150]]></Field>
      <Field name="ScenNo"
type="com.sunopsis.sql.DbInt"><![CDATA[21]]></Field>
      <Field name="ScenTaskNo"
type="com.sunopsis.sql.DbInt"><![CDATA[160]]></Field>
      <Field name="TaskName1" type="java.lang.String"><![CDATA[Insert new
rows]]></Field>
      <Field name="TaskName2" type="java.lang.String"><![CDATA[IKM Oracle
Insert]]></Field>
      <Field name="TaskName3" type="java.lang.String"><![CDATA[Load
UR_IMS_RECORDS]]></Field>
      <Field name="TaskType" type="java.lang.String"><![CDATA[J]]></Field>
</Object>
```

**SQL запит для fraud.ur_ims_records**

```
select
```

```
v.Instance_name,
s.display_name source_name,
a.id alert_id,
a.created_on alert_creation_time,
a.alert_virtual_time alert_virtual_time,
r.name rule_name,
a.number_of_cdrs,
count(1) over (partition by a.id) count_jurs,
min(j.tm_end_time) over(partition by a.id) min_call_end_time,
max(j.tm_end_time) over(partition by a.id) max_call_end_time,
min(j._time_stamp) over(partition by a.id) min_jurt_time,
max(j.jur_time_stamp) over(partition by a.id) max_jurt_time,
min (m.Loading_time) over (partition by a.id) min_loading_time_cdr,
max (m.Loading_time) over (partition by a.id) max_loading_time_cdr,
m.Loading_time,
m.file_name,
p.tm_time_stamp pos_tm_time_stamp
from fraud.alerts a,
        fraud.total_alerted_cdrs j,
        fraud.sources s,
        fraud.rule_sources rs,
        fraud.rules r,
        fraud.pos_prealerts p,
        fraud.ur_ims_records m,
        v$INSTANCE v
where 1=1
and a.created_on between to_date('01/02/2024','dd/mm/yyyy') and
to_date('02/02/2024','dd/mm/yyyy')
and a.id=j.alert_id
and r.id=a.rule_id
and s.id=rs.source_id
and rs.rule_id=r.id
and a.id=p.prealert_id
and s.name='mortadella'
and m.file_name=j.gn_file_name
```

**SQL запит для fraud.ur_nrtrde_records**

```
select
v.Instance_name,
s.display_name source_name,
a.id alert_id,
a.created_on alert_creation_time,
a.alert_virtual_time alert_virtual_time,
```

```
r.name rule_name,
a.number_of_cdrs,
count(1) over (partition by a.id) count_jurs,
min(j.tm_end_time) over(partition by a.id) min_call_end_time,
max(j.tm_end_time) over(partition by a.id) max_call_end_time,
min(j._time_stamp) over(partition by a.id) min_jurt_time,
max(j.jur_time_stamp) over(partition by a.id) max_jurt_time,
min (m.Loading_time) over (partition by a.id) min_loading_time_cdr,
max (m.Loading_time) over (partition by a.id) max_loading_time_cdr,
m.Loading_time,
m.file_name,
p.tm_time_stamp pos_tm_time_stamp
from fraud.alerts a,
        fraud.total_alerted_cdrs j,
        fraud.sources s,
        fraud.rule_sources rs,
        fraud.rules r,
        fraud.pos_prealerts p,
        fraud.ur_nrtrde_records m,
        v$INSTANCE v
where 1=1
and a.created_on between to_date('01/02/2024','dd/mm/yyyy') and
to_date('02/02/2024','dd/mm/yyyy')
and a.id=j.alert_id
and r.id=a.rule_id
and s.id=rs.source_id
and rs.rule_id=r.id
and a.id=p.prealert_id
and s.name='NRTRDE'
and m.file_name=j.gn_file_name
```

**SQL запит для fraud.ur_tap3_records**

```
select
v.Instance_name,
s.display_name source_name,
a.id alert_id,
a.created_on alert_creation_time,
a.alert_virtual_time alert_virtual_time,
r.name rule_name,
a.number_of_cdrs,
count(1) over (partition by a.id) count_jurs,
min(j.tm_end_time) over(partition by a.id) min_call_end_time,
max(j.tm_end_time) over(partition by a.id) max_call_end_time,
```

```
min(j._time_stamp) over(partition by a.id) min_jurt_time,
max(j.jur_time_stamp) over(partition by a.id) max_jurt_time,
min (m.Loading_time) over (partition by a.id) min_loading_time_cdr,
max (m.Loading_time) over (partition by a.id) max_loading_time_cdr,
m.Loading_time,
m.file_name,
p.tm_time_stamp pos_tm_time_stamp
from fraud.alerts a,
        fraud.total_alerted_cdrs j,
        fraud.sources s,
        fraud.rule_sources rs,
        fraud.rules r,
        fraud.pos_prealerts p,
        fraud.ur_tap3_records m,
        v$INSTANCE v
where 1=1
and a.created_on between to_date('01/02/2024','dd/mm/yyyy') and
to_date('02/02/2024','dd/mm/yyyy')
and a.id=j.alert_id
and r.id=a.rule_id
and s.id=rs.source_id
and rs.rule_id=r.id
and a.id=p.prealert_id
and s.name='TAP3'
and m.file_name=j.gn_file_name
```